EPFL

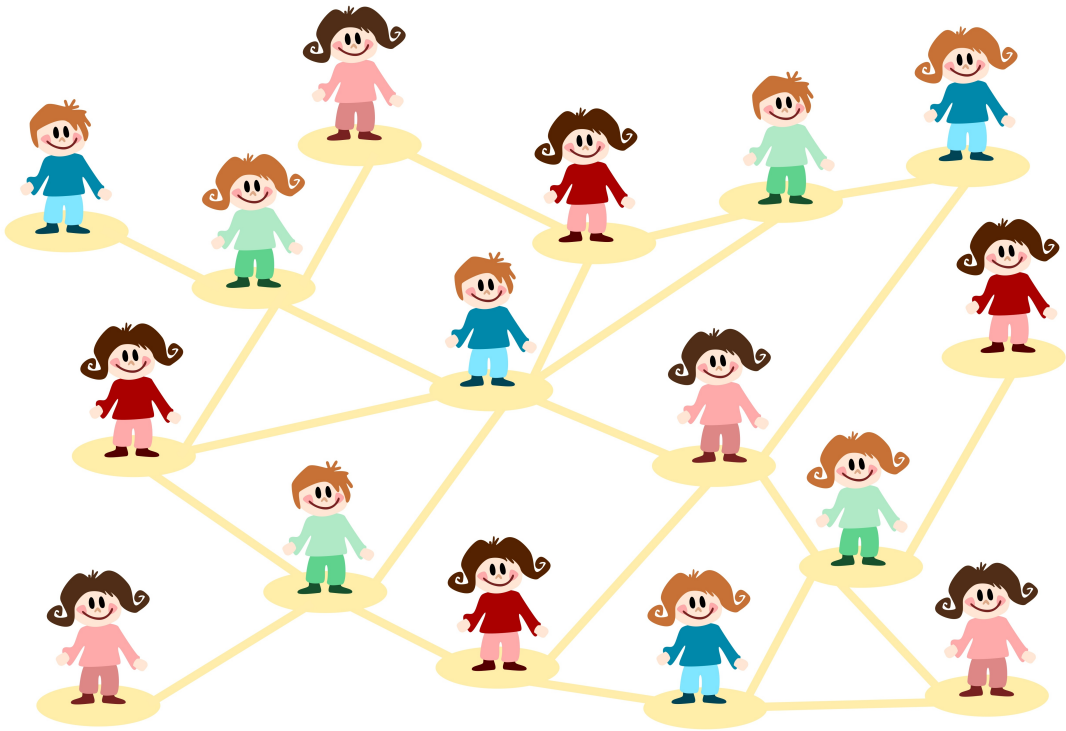
Get More For Less In Decentralized Learning Systems

Akash Dhasade Anne-Marie Kermarrec Rafael Pires **Rishi Sharma** Milos Vujasinovic Jeffrey Wigger

Scalable Computing Systems Laboratory
EPFL

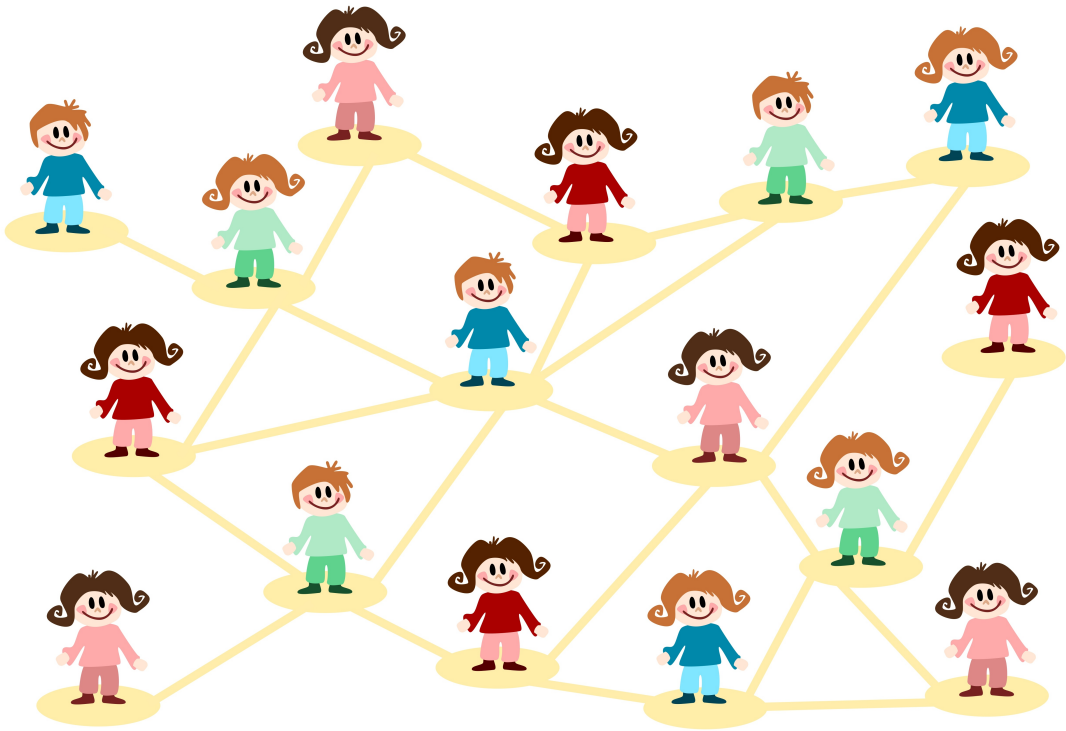
20.07.2023

Decentralized Learning



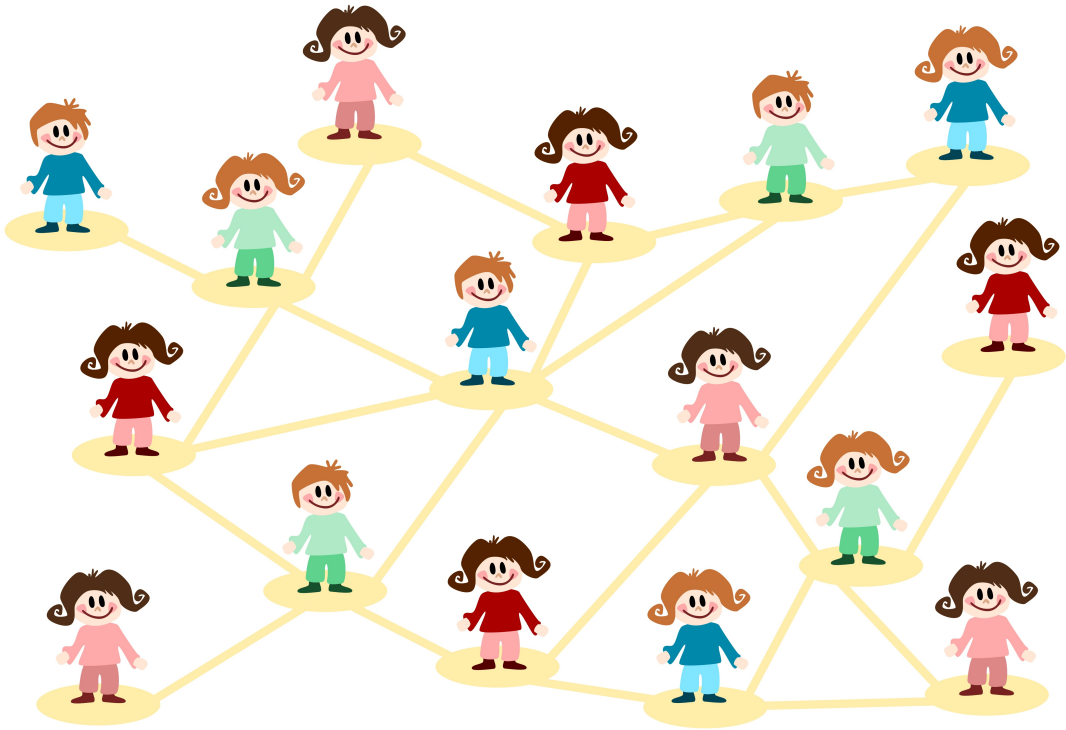
Decentralized Learning

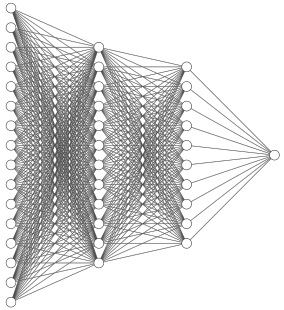
Local data



Decentralized Learning

Local data

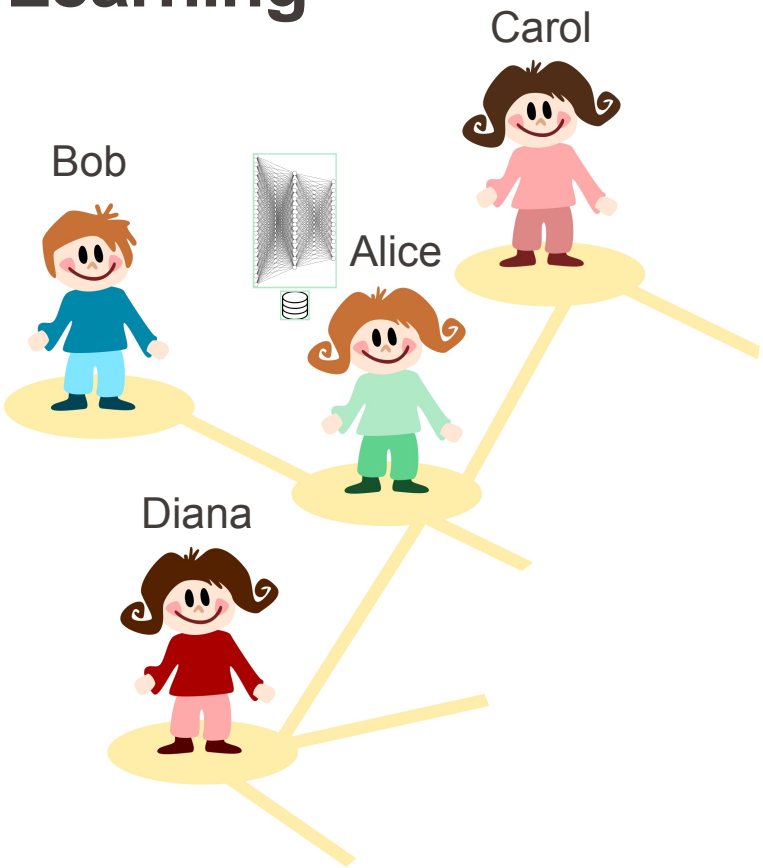




Model architecture

Decentralized Learning

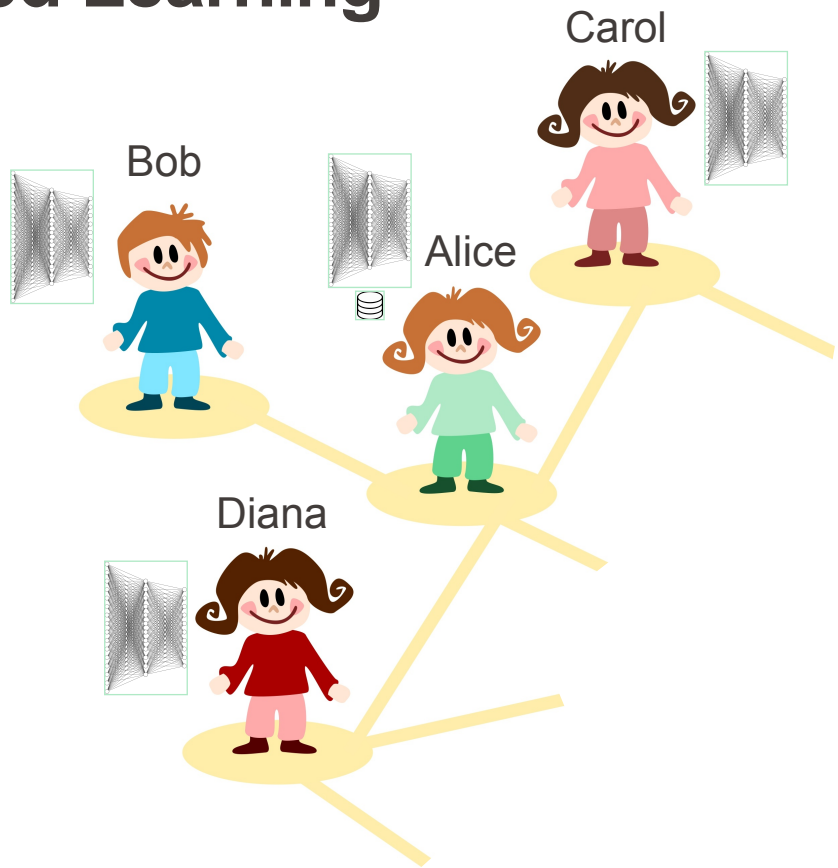
Train locally



Decentralized Learning

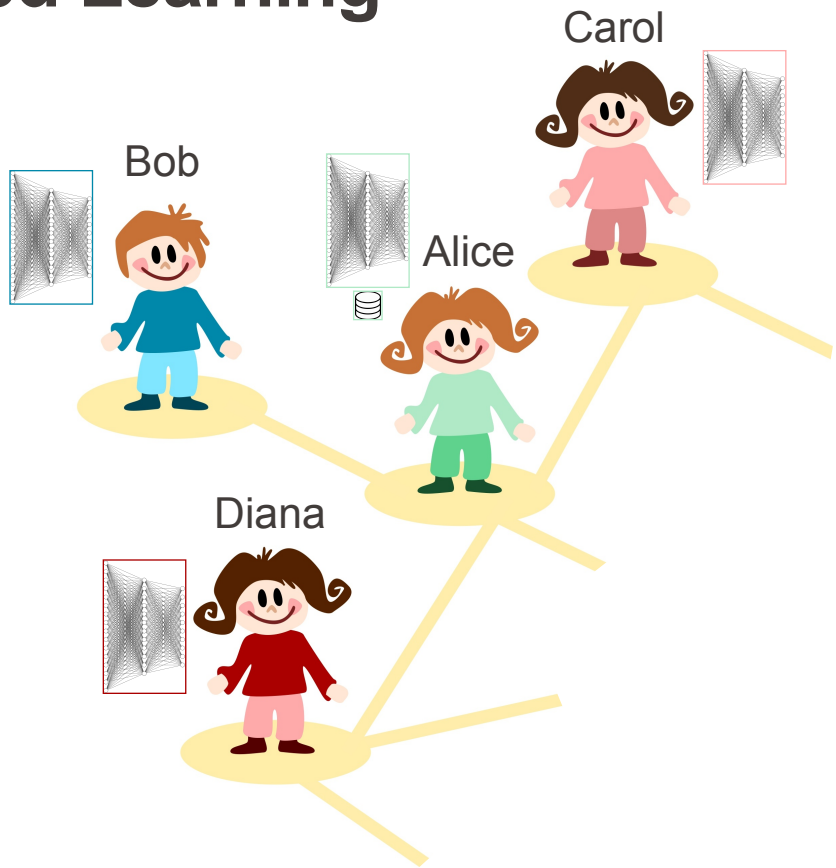
Train locally

Send model



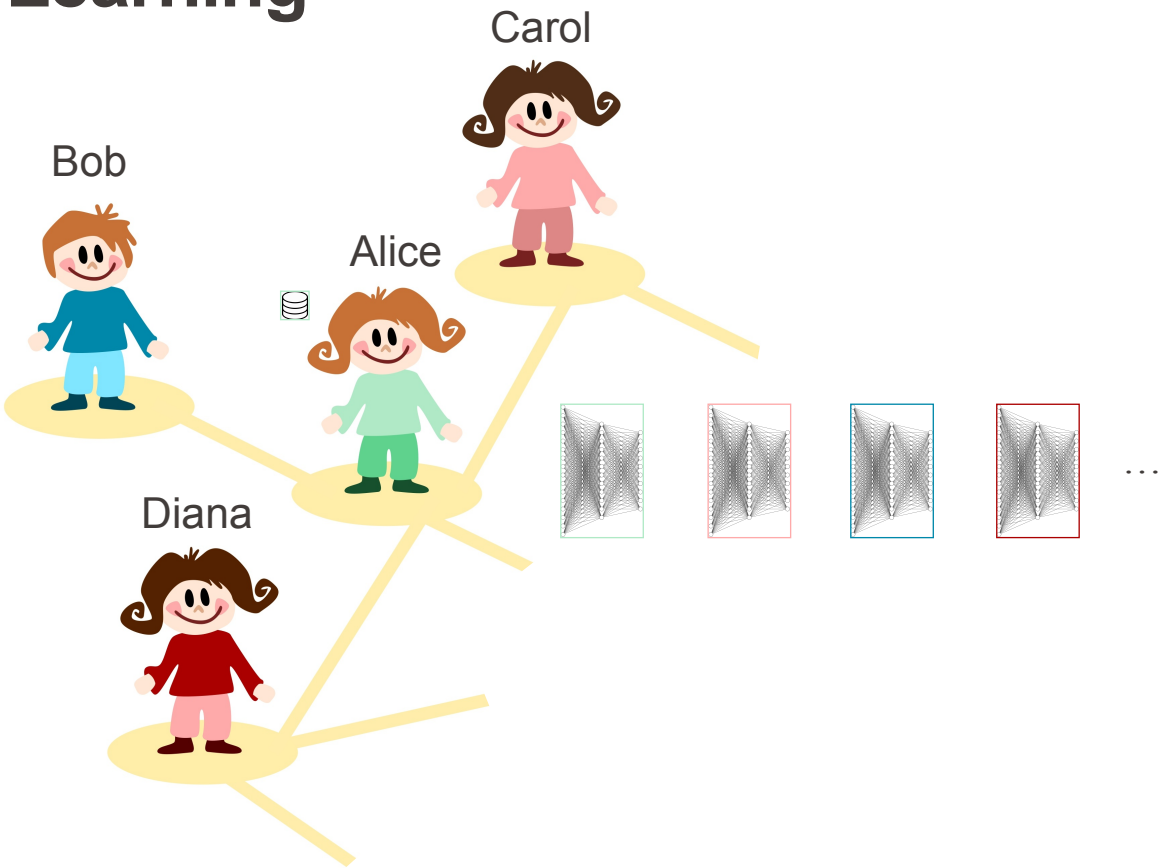
Decentralized Learning

Receive models



Decentralized Learning

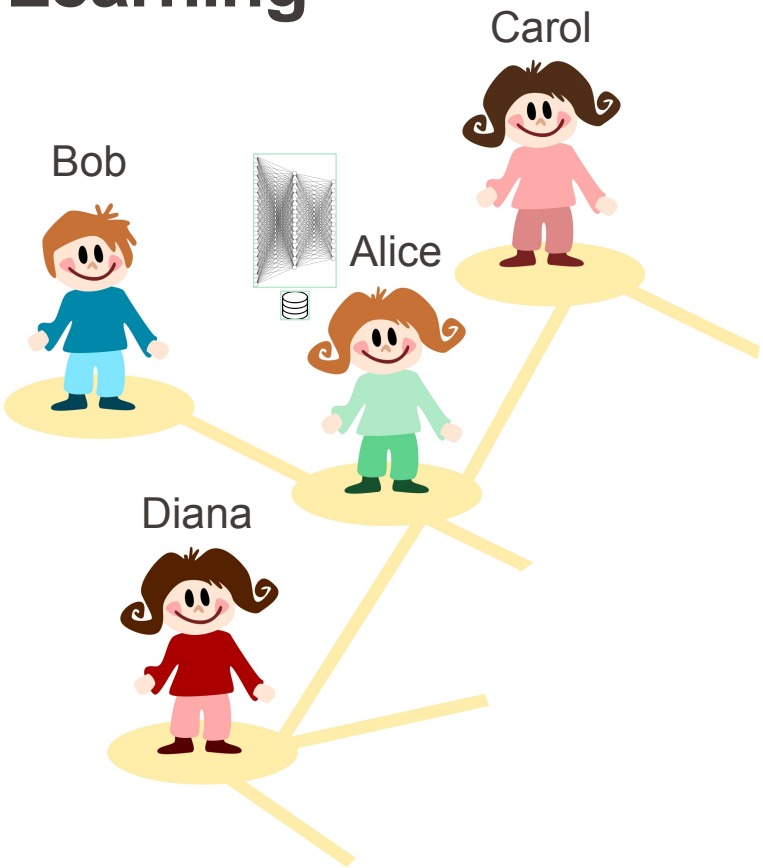
Receive models



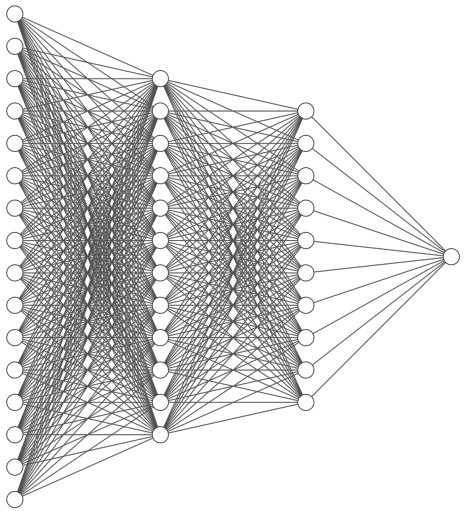
Decentralized Learning

Receive models

Aggregate

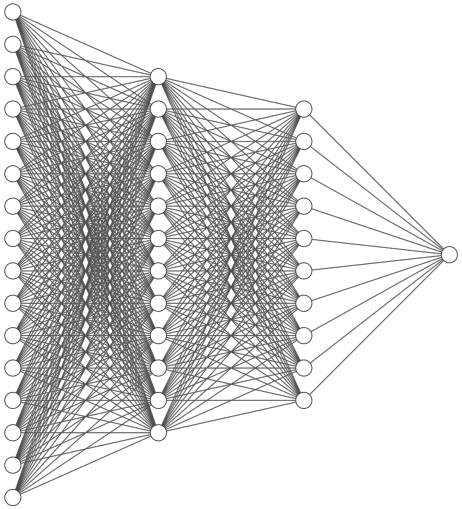


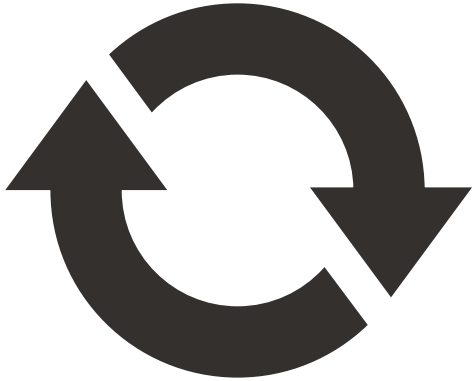
The Problem



Large models

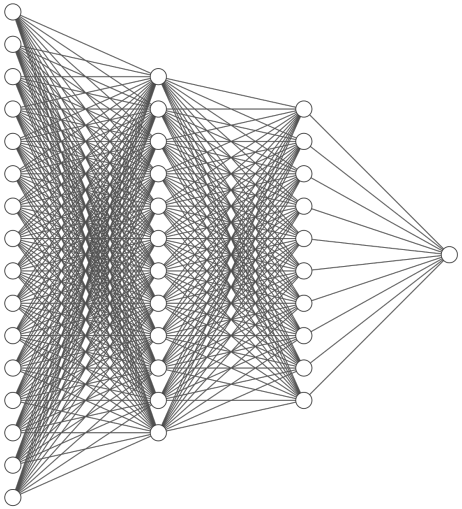
The Problem

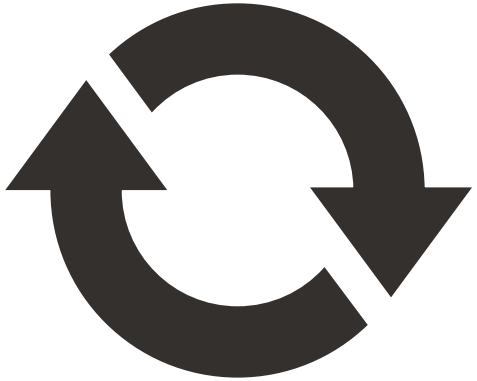


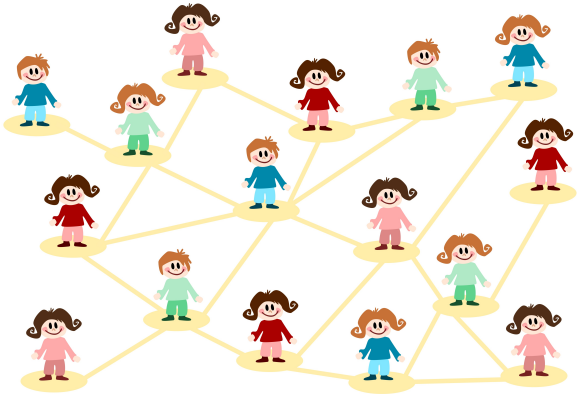


Large models * Many rounds

The Problem

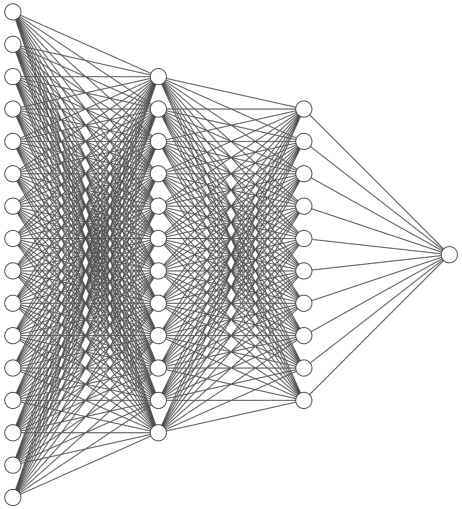


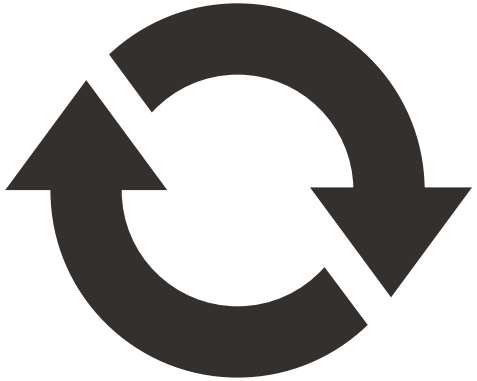


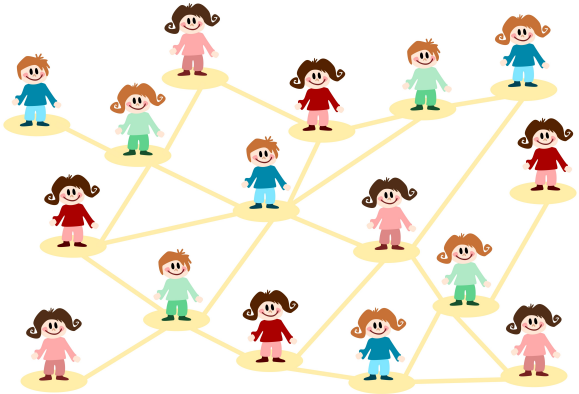


Large models * Many rounds * Multiple peers

The Problem







Large models * Many rounds * Multiple peers

TBs of data transferred

Popular Design: Sparsification

Original parameter vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Popular Design: Sparsification

Original parameter vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Sparsified vector to send

		0.09	0.11	-0.23		0.12				0.98	0.02

Popular Design: Sparsification

Original parameter vector	0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02
Parameter importance	4.6	2.2	**5.2**	**8.8**	**5.3**	2.4	**6.5**	5.1	1.2	2.2	**7.1**	**6.9**

Sparsified vector to send

	0.09	0.11	-0.23		0.12		0.98	0.02

Popular Design: Sparsification

Original parameter vector	0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02
Parameter importance	4.6	2.2	**5.2**	**8.8**	**5.3**	2.4	**6.5**	5.1	1.2	2.2	**7.1**	**6.9**

Random sampling

TopK change

Sparsified vector to send

	0.09	0.11	-0.23		0.12		0.98	0.02

Problem with Sparsification

Original parameter vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Sparsified vector to send

		0.09	0.11	-0.23		0.12				0.98	0.02

Information Loss

Large number of nodes

Non-IID data distributions

Just What is Needed Sharing (JWINS)

Sparsification ✓

Information Loss ✗

Just What is Needed Sharing (JWINS)

Sparsification ✓

Information Loss ✗

Parameter ranking	Parameter selection	Overhead optimization

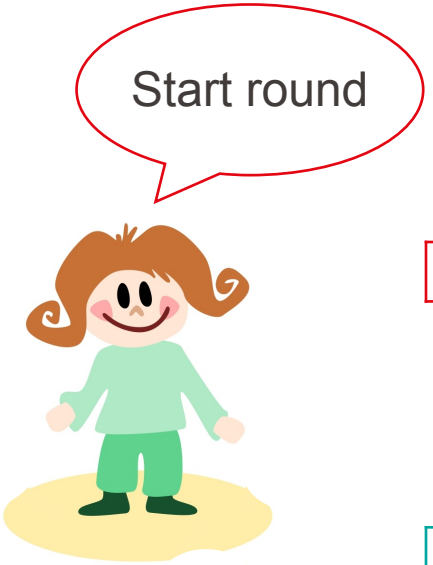
Just What is Needed Sharing (JWINS)

Sparsification ✓

Information Loss ✗

Parameter ranking	Parameter selection	Overhead optimization

Parameter Ranking



Alice

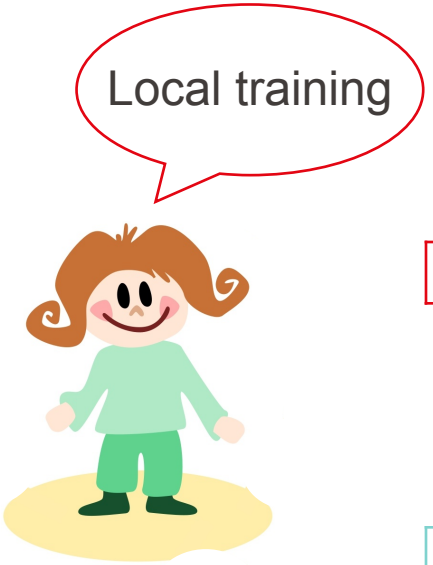
0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Initial Model
T = t

0	2.2	3.4	0.2	4.5	2.4	0	5.1	1.2	0	0.1	0.1

Importance scores
T = t - 1

Parameter Ranking



Alice

1.92	2.23	4.44	1.12	0.28	0.95	0.12	2.52	1.99	2.33	0.09	0.99

Absolute change
T = t

0	2.2	3.4	0.2	4.5	2.4	0	5.1	1.2	0	0.1	0.1

Importance scores
T = t - 1

Parameter Ranking

Local training

Alice

Like a flattened image

1.92	2.23	4.44	1.12	0.28	0.95	0.12	2.52	1.99	2.33	0.09	0.99

Absolute change
T = t

0	2.2	3.4	0.2	4.5	2.4	0	5.1	1.2	0	0.1	0.1

Importance scores
T = t - 1

Wavelet Transform

By Alessio Damato - Own work

Parameter Ranking

Alice

1.92	2.23	4.44	1.12	0.28	0.95	0.12	2.52	1.99	2.33	0.09	0.99

Absolute change
T = t

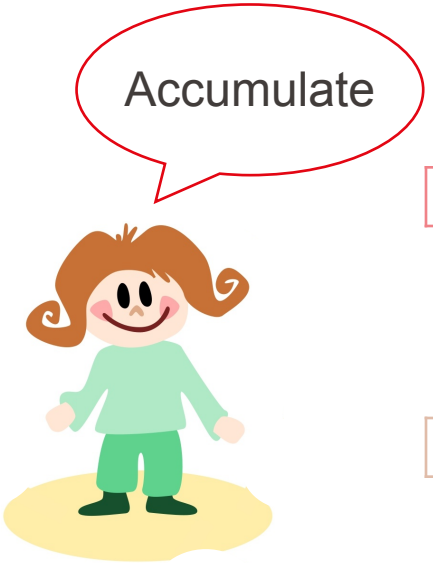
2.09	1.95	1.11	0.11	2.93	5.55	4.00	4.42	5.54	2.43	2.22	5.99

Change wavelets
T = t

0	2.2	3.4	0.2	4.5	2.4	0	5.1	1.2	0	0.1	0.1

Importance scores
T = t - 1

Parameter Ranking



Alice

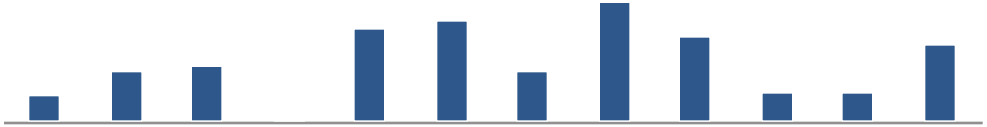
1.92	2.23	4.44	1.12	0.28	0.95	0.12	2.52	1.99	2.33	0.09	0.99

Absolute change
$T = t$

2.09	1.95	1.11	0.11	2.93	5.55	4.00	4.42	5.54	2.43	2.22	5.99

Change wavelets
$T = t$

2.09	4.15	4.51	0.31	7.43	7.95	4	9.52	6.74	2.43	2.32	6.09



Importance scores
$T = t$

Parameter Ranking

Alice

1.92	2.23	4.44	1.12	0.28	0.95	0.12	2.52	1.99	2.33	0.09	0.99

Absolute change
$T = t$

2.09	1.95	1.11	0.11	2.93	5.55	4.00	4.42	5.54	2.43	2.22	5.99

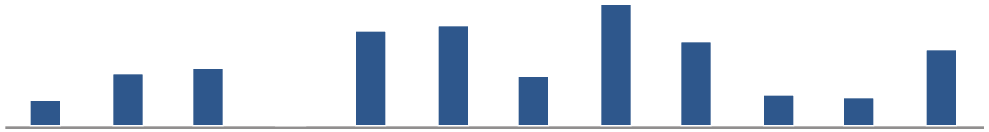
Change wavelets
$T = t$

2.09	4.15	**0**	0.31	**0**	**0**	4	**0**	**0**	2.43	2.32	**0**

Importance scores
$T = t$

Just What is Needed Sharing (JWINS)

2.09	4.15	**4.51**	0.31	**7.43**	**7.95**	4	**9.52**	**6.74**	2.43	2.32	**6.09**



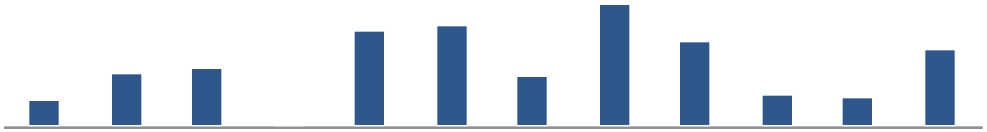
Importance scores
$T = t$

Parameter ranking → **Parameter selection** | Overhead optimization

Parameter Selection

Alice

2.09	4.15	4.51	0.31	**7.43**	**7.95**	4	**9.52**	6.74	2.43	2.32	6.09



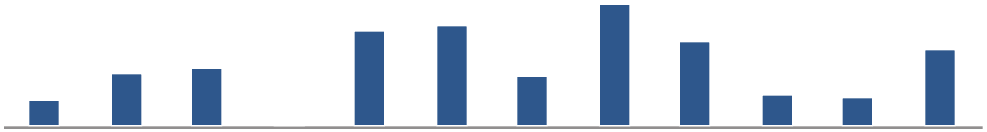
Importance scores
$T = t$

Parameter Selection

50 rounds to share

Alice

2.09	4.15	4.51	0.31	**7.43**	**7.95**	4	**9.52**	6.74	2.43	2.32	6.09



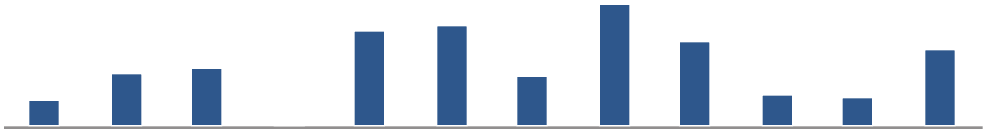
Importance scores
$T = t$

Parameter Selection

50 rounds to share

Alice

2.09	4.15	4.51	0.31	**7.43**	**7.95**	4	**9.52**	6.74	2.43	2.32	6.09



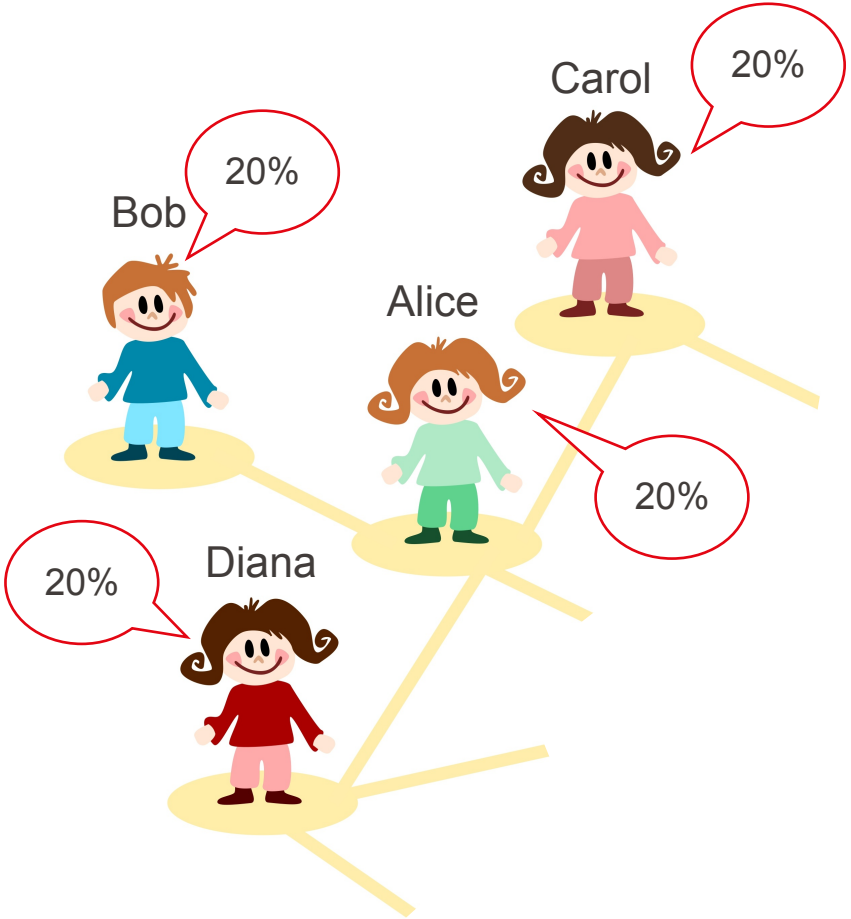
Importance scores
$T = t$

Accumulation helps, but is SLOW!

Bypass accumulation!

Parameter Selection

Communication Goal: 20%

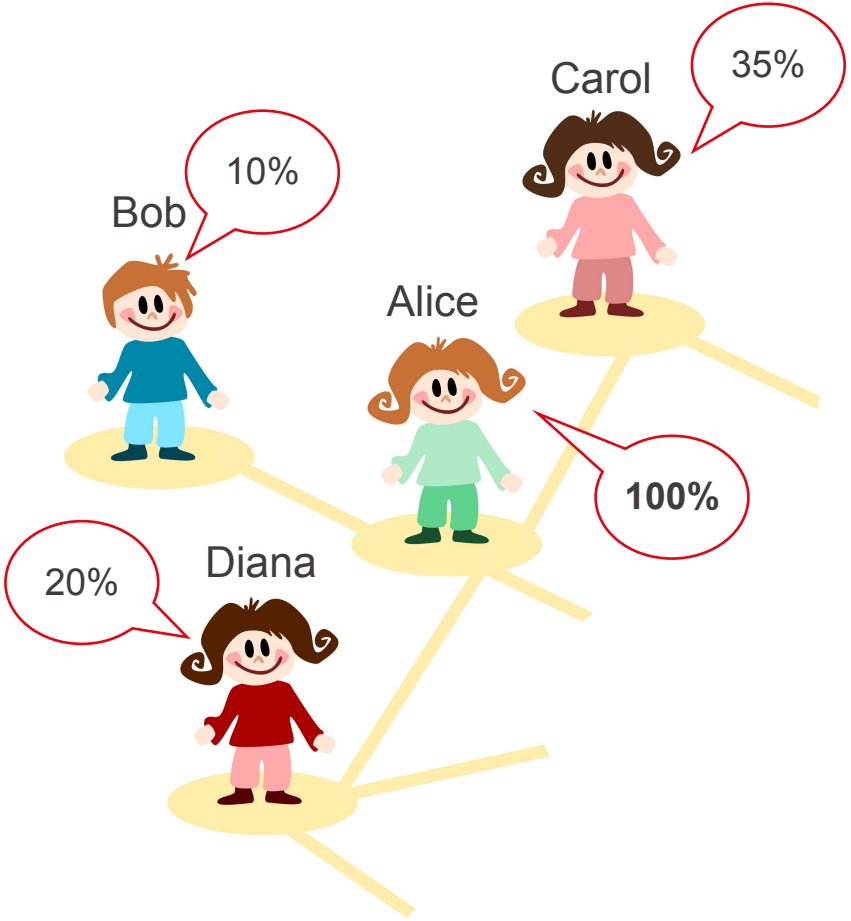


Sparsification

Parameter Selection

Communication Goal: 20%

Randomized Cut-off

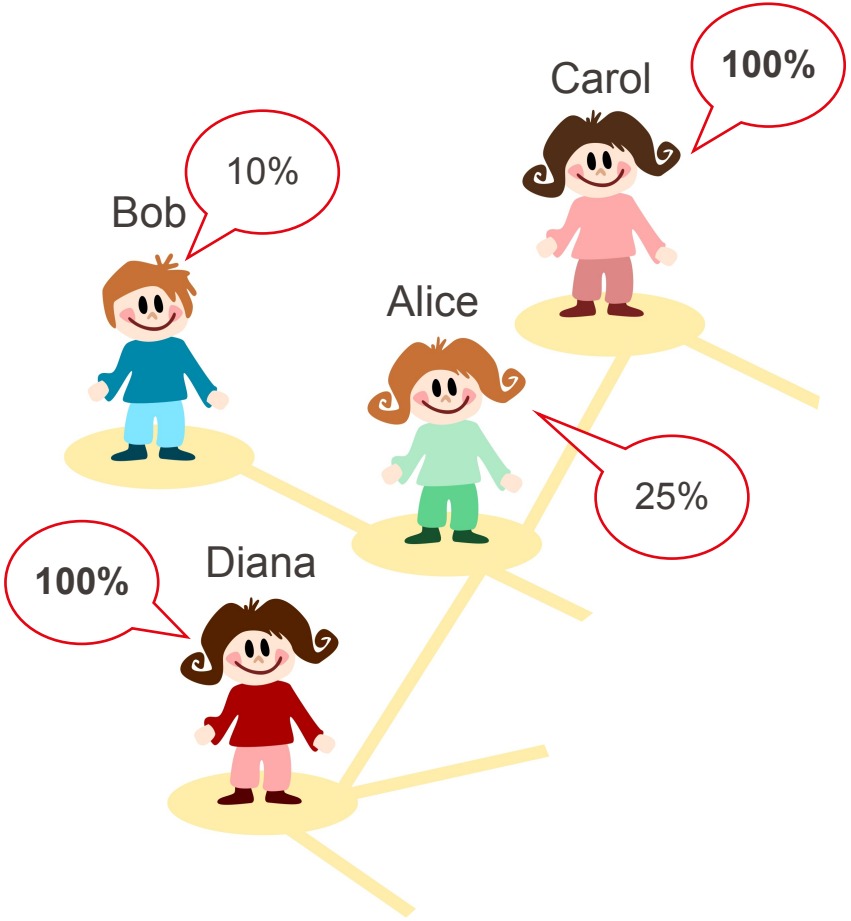


JWINS

Parameter Selection

Communication Goal: 20%

Randomized Cut-off

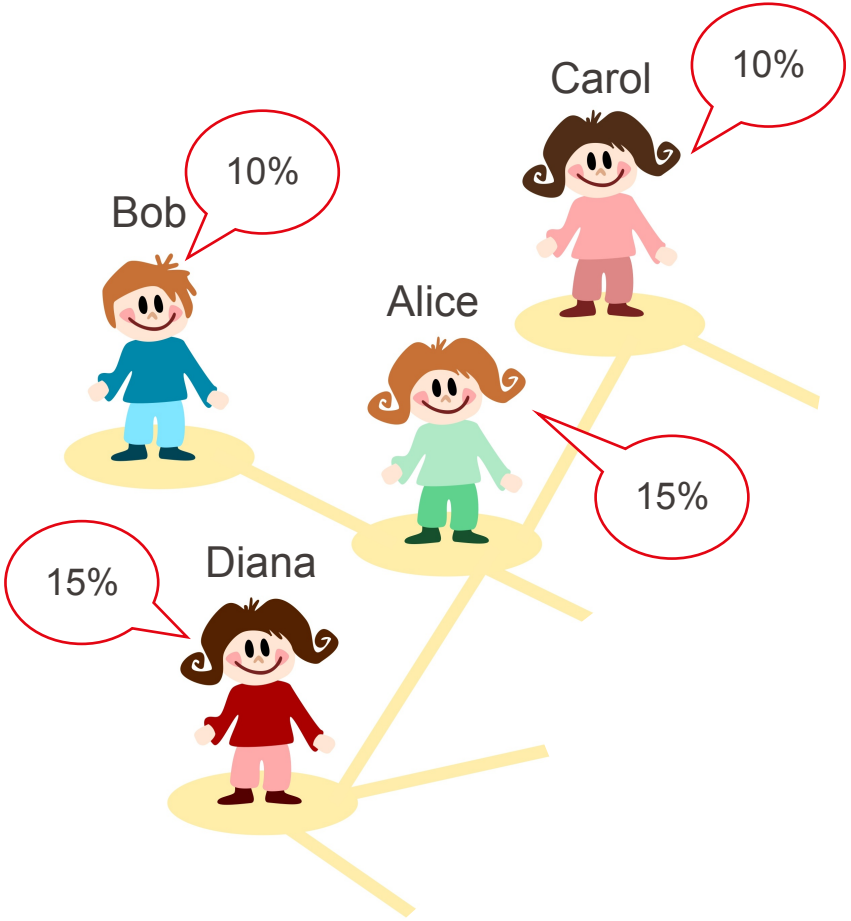


JWINS

Parameter Selection

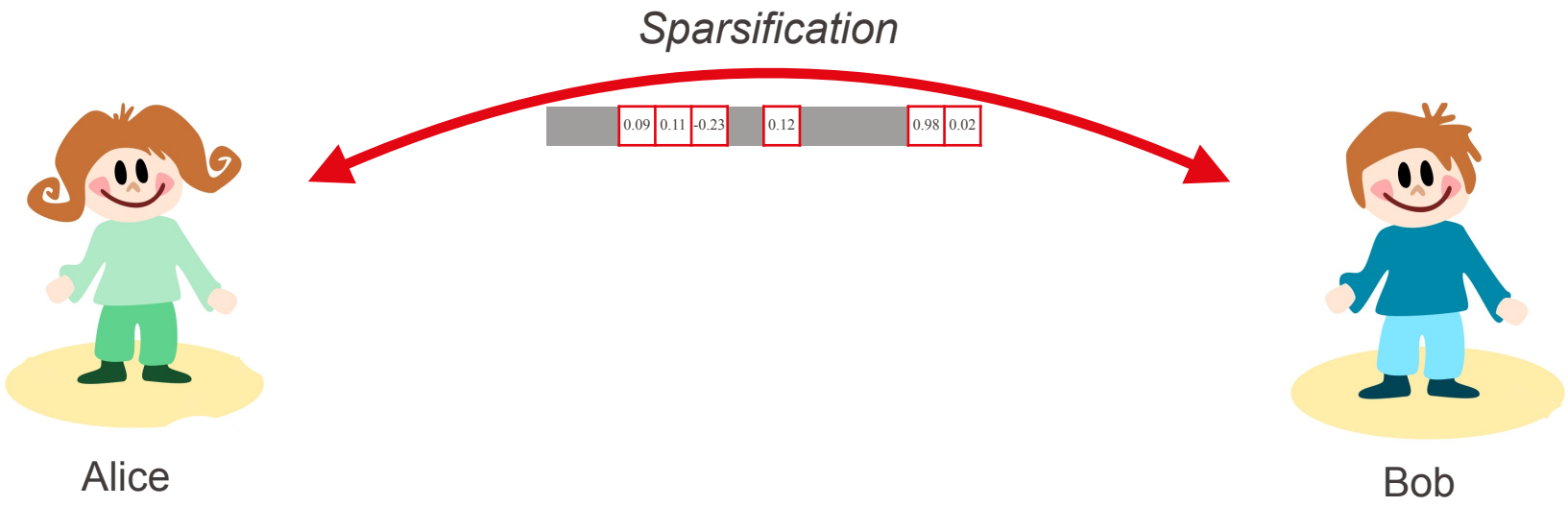
Communication Goal: 20%

Randomized Cut-off



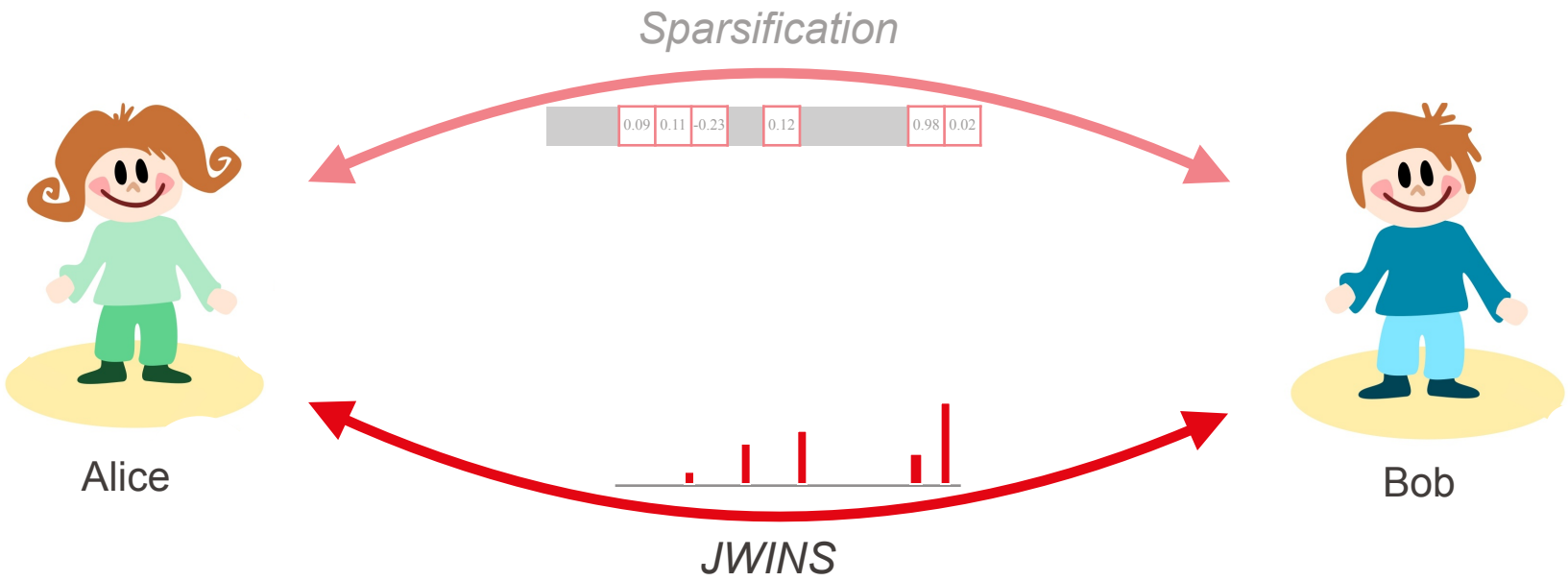
JWINS

Just What is Needed Sharing (JWINS)



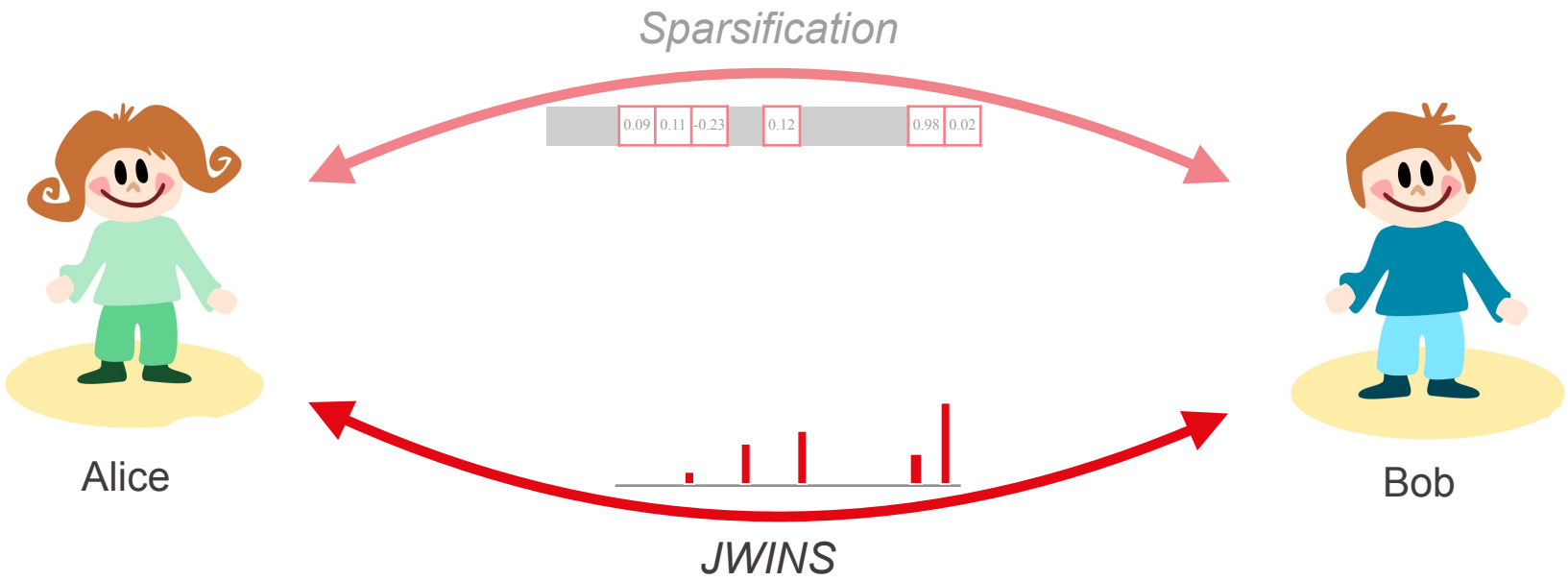
Parameter ranking	**Parameter selection**	Overhead optimization

Just What is Needed Sharing (JWINS)



Parameter ranking	**Parameter selection**	Overhead optimization

Just What is Needed Sharing (JWINS)



Parameter ranking | Parameter selection → **Overhead optimization**

Evaluation

Task	Dataset	Model	Total Parameters
Image Classification	CelebA	4 Conv2D Layers	30 242
Image Classification	CIFAR-10	GN-LeNet	89 834
Image Classification	FEMNIST	2 Conv2D Layers	1 690 046
Recommendation	MovieLens	Matrix Factorization	103 340
Next Character Prediction	Shakespeare (small)	Stacked LSTM	2 683 708

Evaluation

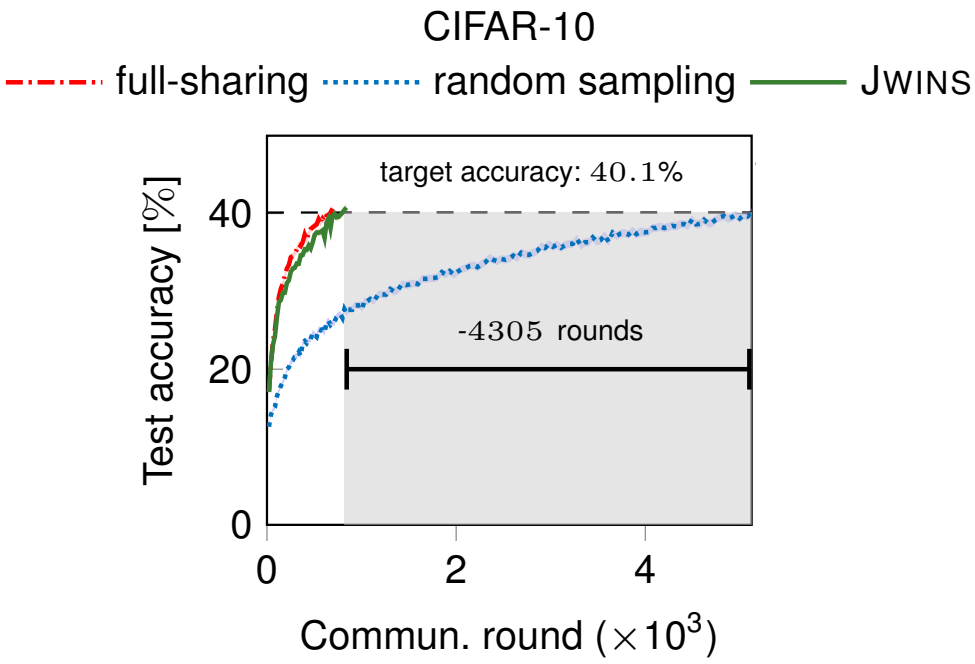
Task	Dataset	Model	Total Parameters
Image Classification	CelebA	4 Conv2D Layers	30 242
Image Classification	CIFAR-10	GN-LeNet	89 834
Image Classification	FEMNIST	2 Conv2D Layers	1 690 046
Recommendation	MovieLens	Matrix Factorization	103 340
Next Character Prediction	Shakespeare (small)	Stacked LSTM	2 683 708

Topology: **96 nodes** | **4-regular random**
Data Distribution: **Non-IID**
Metric: **Top-1 Accuracy** | **Bytes sent**
Baselines: **Full-sharing** | **Random sampling** | **Choco-SGD**

Evaluation: Rounds to Accuracy

Communication budget: 37%

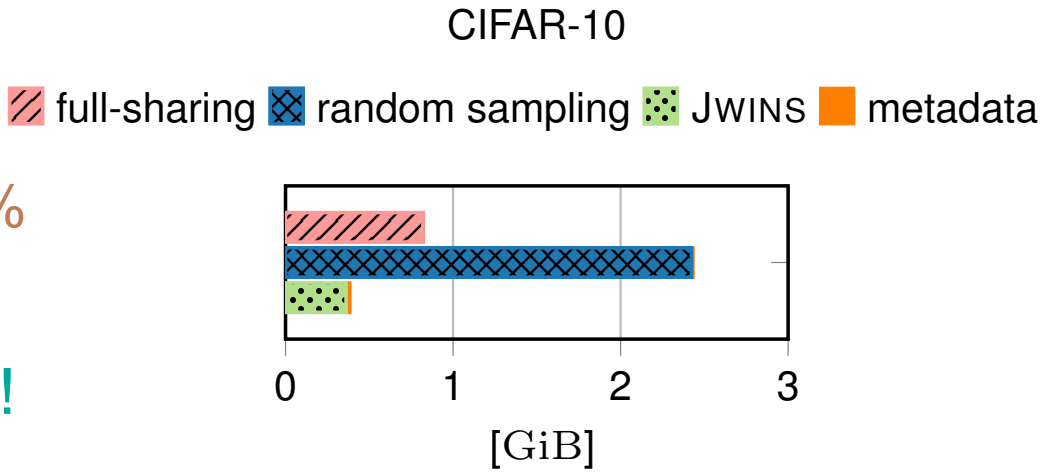
4x faster!



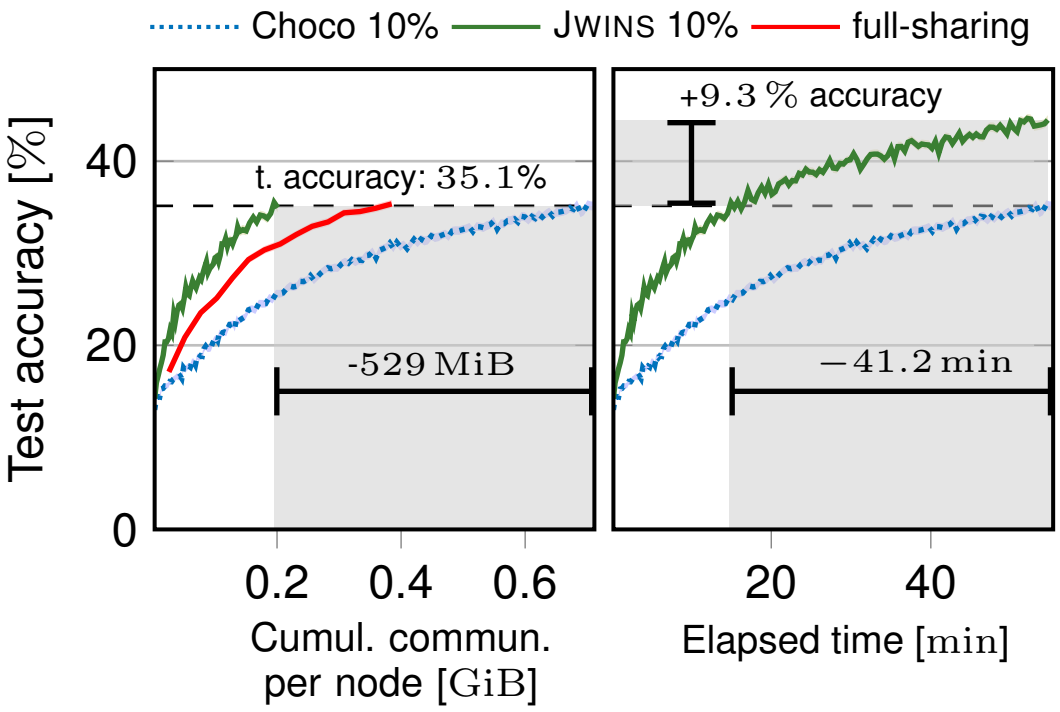
Evaluation: Comm. Cost to Accuracy

Communication budget: 37%

More than 4x communication efficiency!



Evaluation: Choco-SGD



3.9x faster!

Consistent performance at lower budgets.

Key Takeaways

- ✳ Decentralized learning is communication hungry
- ✳ Sparsification helps, but fails at scale in non-IID
- ✓ JWINS is 1.5x - 4x more communication efficient
- ✓ JWINS reaches 2% - 15% better accuracy

Get More for Less in Decentralized Learning Systems

Akash Dhasade*, Anne-Marie Kermarrec*, Rafael Pires*, Rishi Sharma*†, Milos Vujasinovic* and Jeffrey Wigger‡
*EPFL, Switzerland, first.last@epfl.ch
†Corresponding author. ‡Unaffiliated.

***Abstract*—Decentralized learning (DL) systems have been gaining popularity because they avoid raw data sharing by communicating only model parameters, hence preserving data confidentiality. However, the large size of deep neural networks poses a significant challenge for decentralized training, since each node needs to exchange gigabytes of data, overloading the network. In this paper, we address this challenge with JWINS, a communication-efficient and fully decentralized learning system that shares only a subset of parameters through sparsification. JWINS uses wavelet transform to limit the information loss due to sparsification and a randomized communication cut-off that reduces communication usage without damaging the performance of trained models. We demonstrate empirically with 96 DL nodes on non-IID datasets that JWINS can achieve similar accuracies to full-sharing DL while sending up to 64% fewer bytes. Additionally, on low communication budgets, JWINS outperforms the state-of-the-art communication-efficient DL algorithm CHOCO-SGD by up to 4× in terms of network savings and time.**

***Index Terms*—sparsification, compression, communication efficiency, decentralized, P2P, machine learning**

I. Introduction

Deep learning algorithms have significantly improved artificial intelligence tasks, such as image classification [1], speech recognition [2], and text detection [3]. Enormous volumes of data are produced daily on smartphones and edge devices, which greatly enhance the performance of deep learning models. In addition to facing challenges in efficiently handling the massive size of users' data, the centralized data collection also poses a considerable privacy threat, since personal data can be sensitive to users and its exposure represents a significant concern.

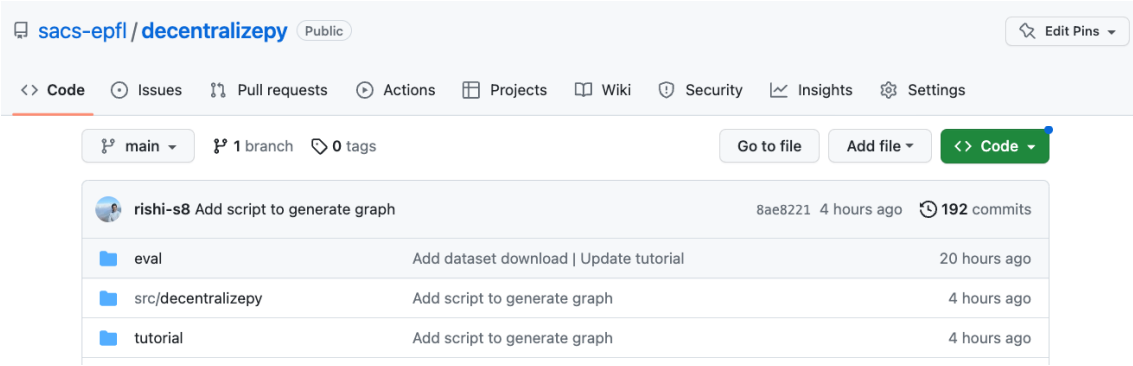
Decentralized learning (DL) has emerged as an attractive alternative for training models on decentralized data, where nodes learn by exchanging models among themselves instead of sharing data with large service providers. This approach provides additional privacy [4]–[7] and obviates the need for complex data management in data centers. Furthermore, DL systems are also popular for their enhanced scalability compared to centralized systems, where the server can become the bottleneck of the system [4]. Nodes in DL are typically devices at the edge of the network, with limited bandwidth, and connected according to a communication topology. While powerful in computational capabilities, these devices must exchange machine learning (ML) models under constrained communication budgets. As a consequence, minimizing the number of bytes transferred when performing DL training has been a major focus of research [6]–[9].

An important technique to minimize the communication during training is *sparsification* or *partial sharing*, which limits the model transfer to a small fraction of *important* parameters [10]–[13]. For sparsification to be effective, it is crucial to select these important parameters with care, ensuring that the resulting global model remains unbiased and maintains the expected accuracy. The rest of the parameters remain local in favor of communication efficiency, thus limiting the information exchanged between nodes. Considering that the data distribution across nodes is often non independent and identically distributed (non-IID), the withheld information due to sparsification can have a detrimental effect on model utility [14]. While there have been in-depth theoretical studies in sparsification-based communication efficiency for DL systems [6], [7], empirical studies with hundreds of nodes and challenging non-IID datasets have been fairly limited. Most settings either use a central coordinator or all-to-all communication [10], [14], [15], which are both unpractical in decentralized systems.

In this paper, we present JWINS (*just what is needed sharing*), a system that performs decentralized training under limited communication budgets while retaining good model performance. The novelty of JWINS lies in the way JWINS addresses the critical issue of choosing important parameters by (1) ranking the parameters and averaging the sparse model vectors in the wavelet-frequency domain, and (2) choosing the size of the sparse model vector using a randomized cut-off scheme which respects the overall communication budget. Since each wavelet coefficient contains information about a subset of the original parameters, JWINS can pack more information in the sparse vectors compared to standard sparsification techniques that rely only on the magnitude of parameter updates. The parameters are both shared and averaged as wavelet coefficients, thus limiting the loss of information due to parameter sparsification in each round. Finally, with the randomized cut-off scheme, each node gets the chance to share the full model parameters every few rounds without violating the communication budget of the entire training process. This shift of approach to the wavelet-frequency domain results in scalable and performant sparsification.

We evaluated JWINS in non-IID data distributions over the extensively used datasets of CIFAR-10, MovieLens, and

JWINS: Decentralized learning made communication efficient!



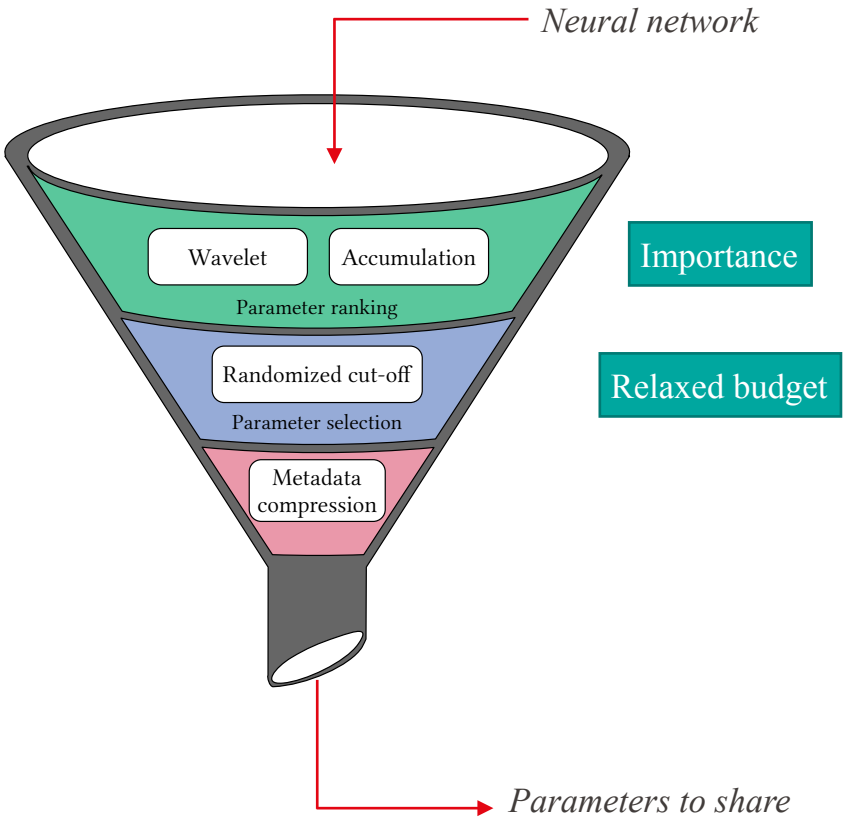
https://github.com/sacs-epfl/decentralizepy

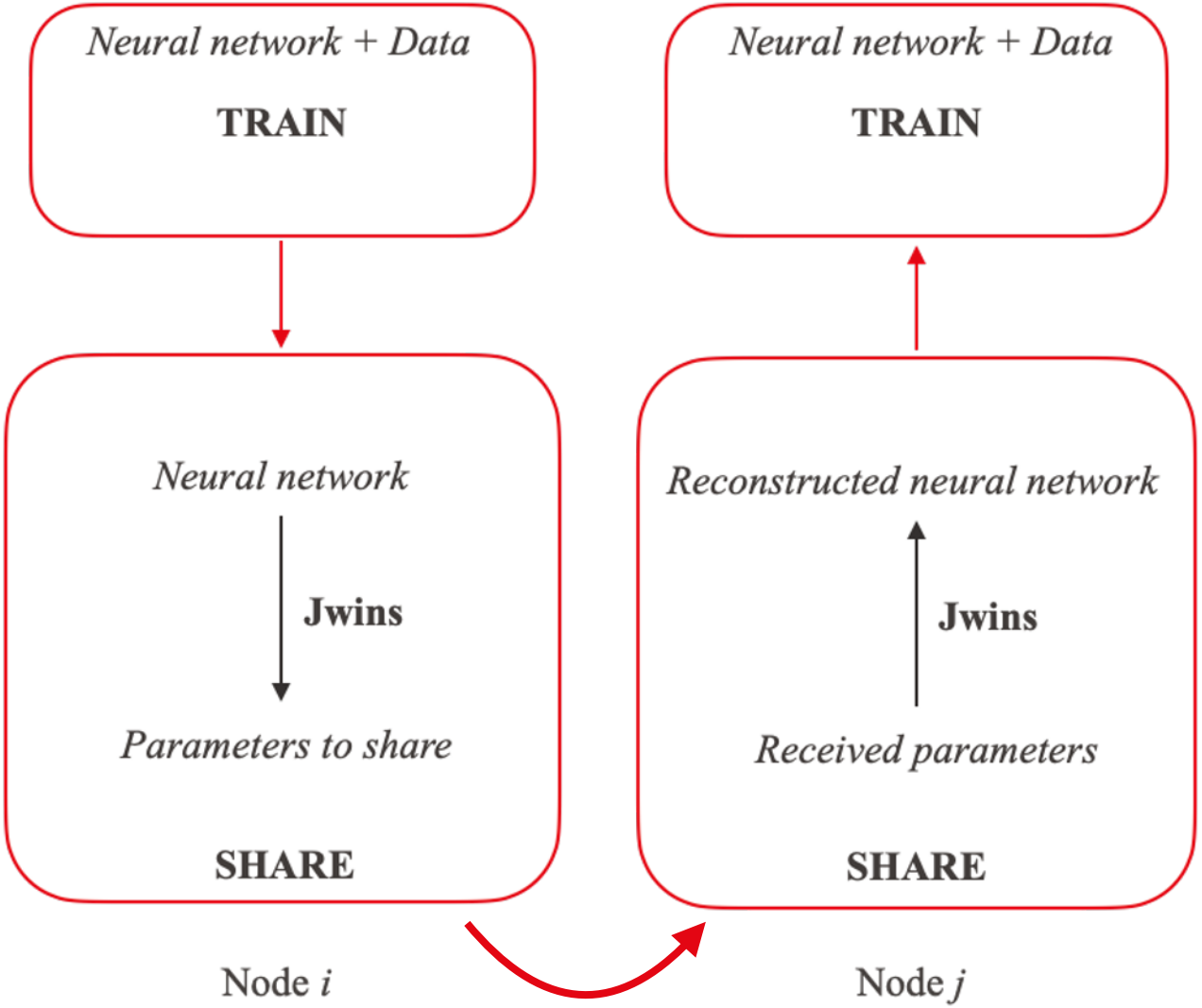
rishi-s8.github.io

rishi.sharma@epfl.ch

Backup Slides

Just What is Needed Sharing (JWINS)

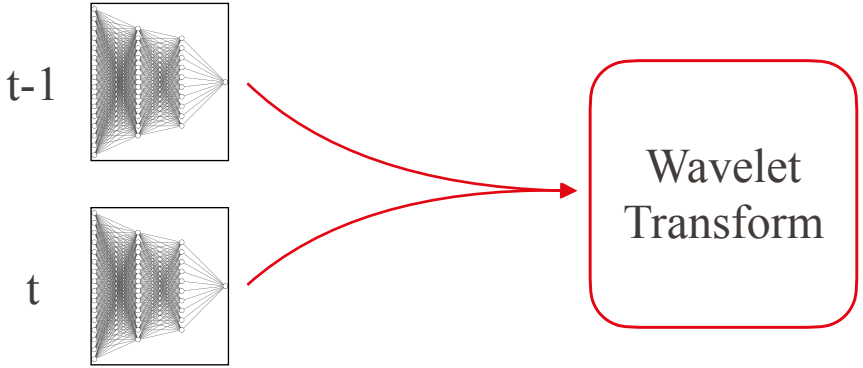



Neural network + Data
TRAIN
Neural network
Jwins
Parameters to share
SHARE
Node i
Neural network + Data
TRAIN
Reconstructed neural network
Jwins
Received parameters
SHARE
Node j

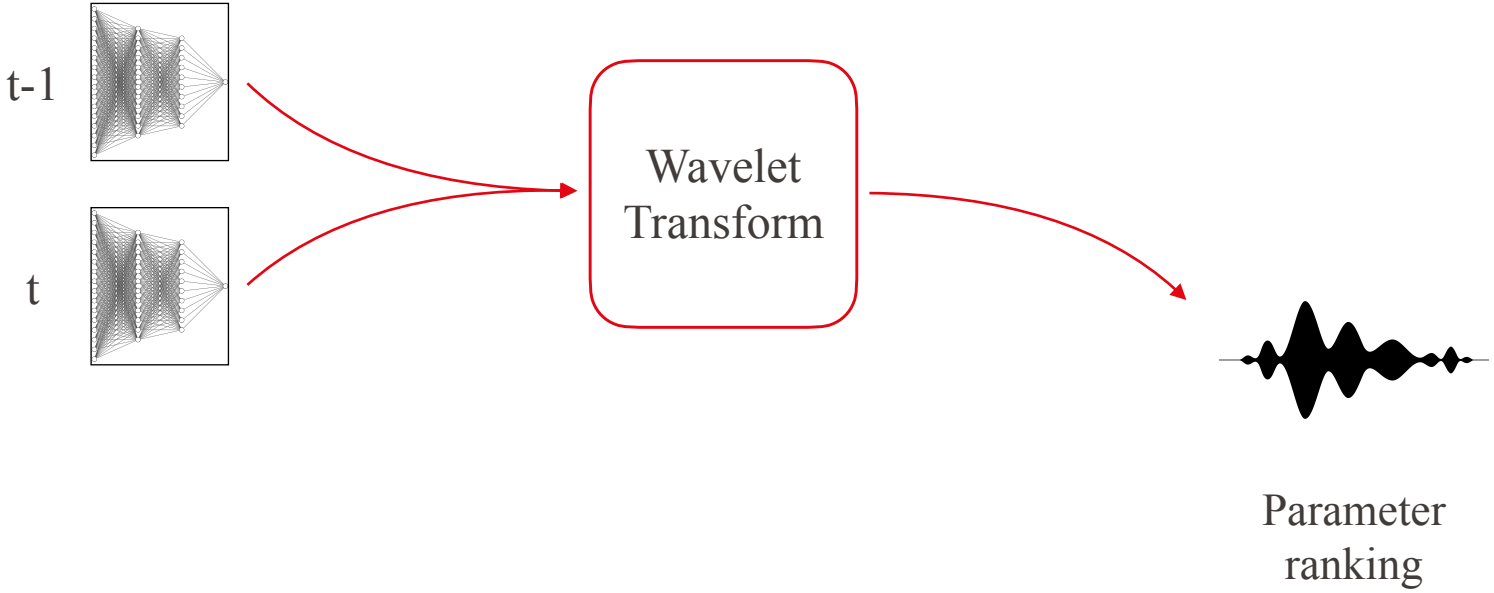
Parameter Ranking

Wavelet
Transform

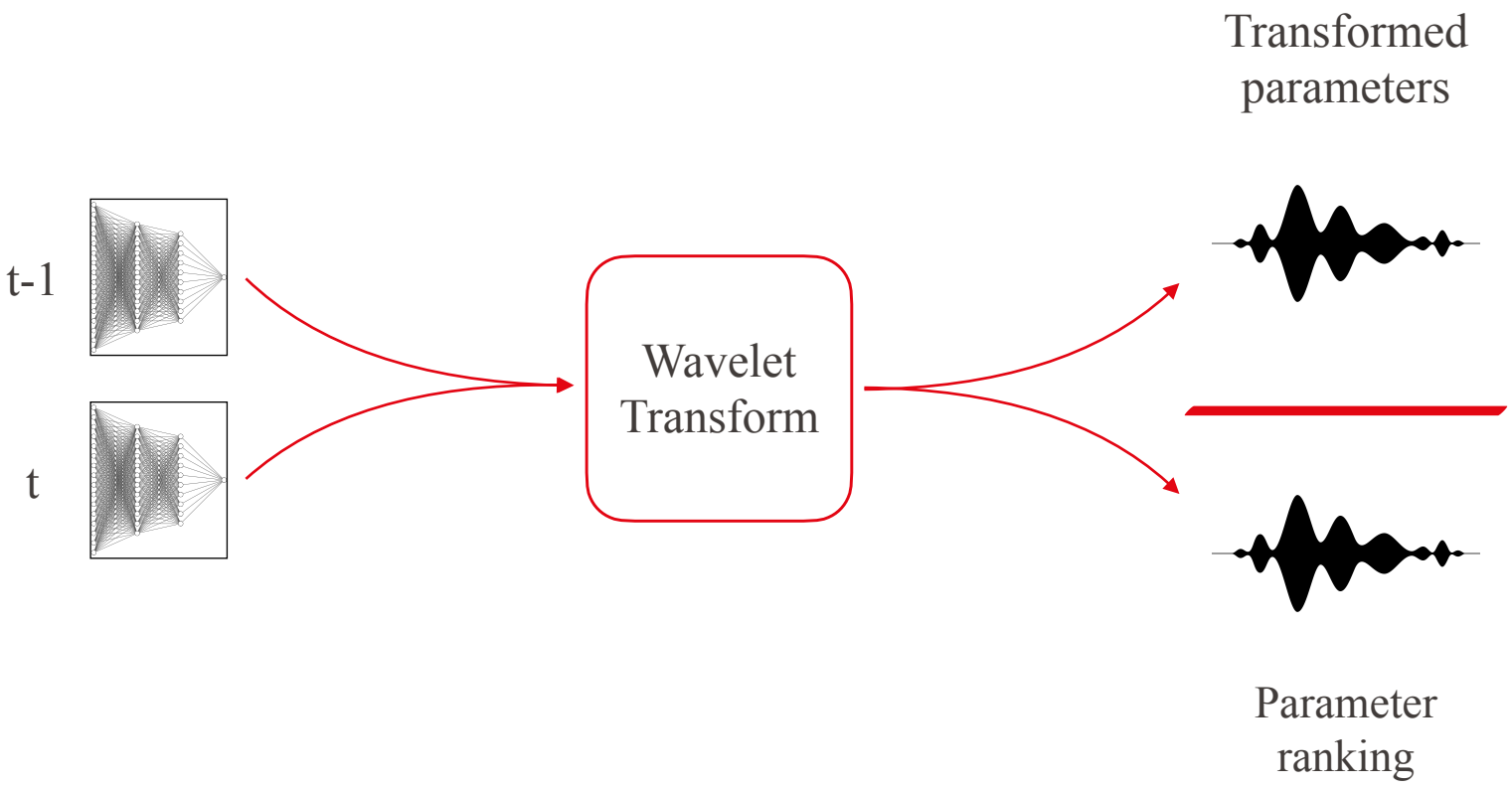
Parameter Ranking



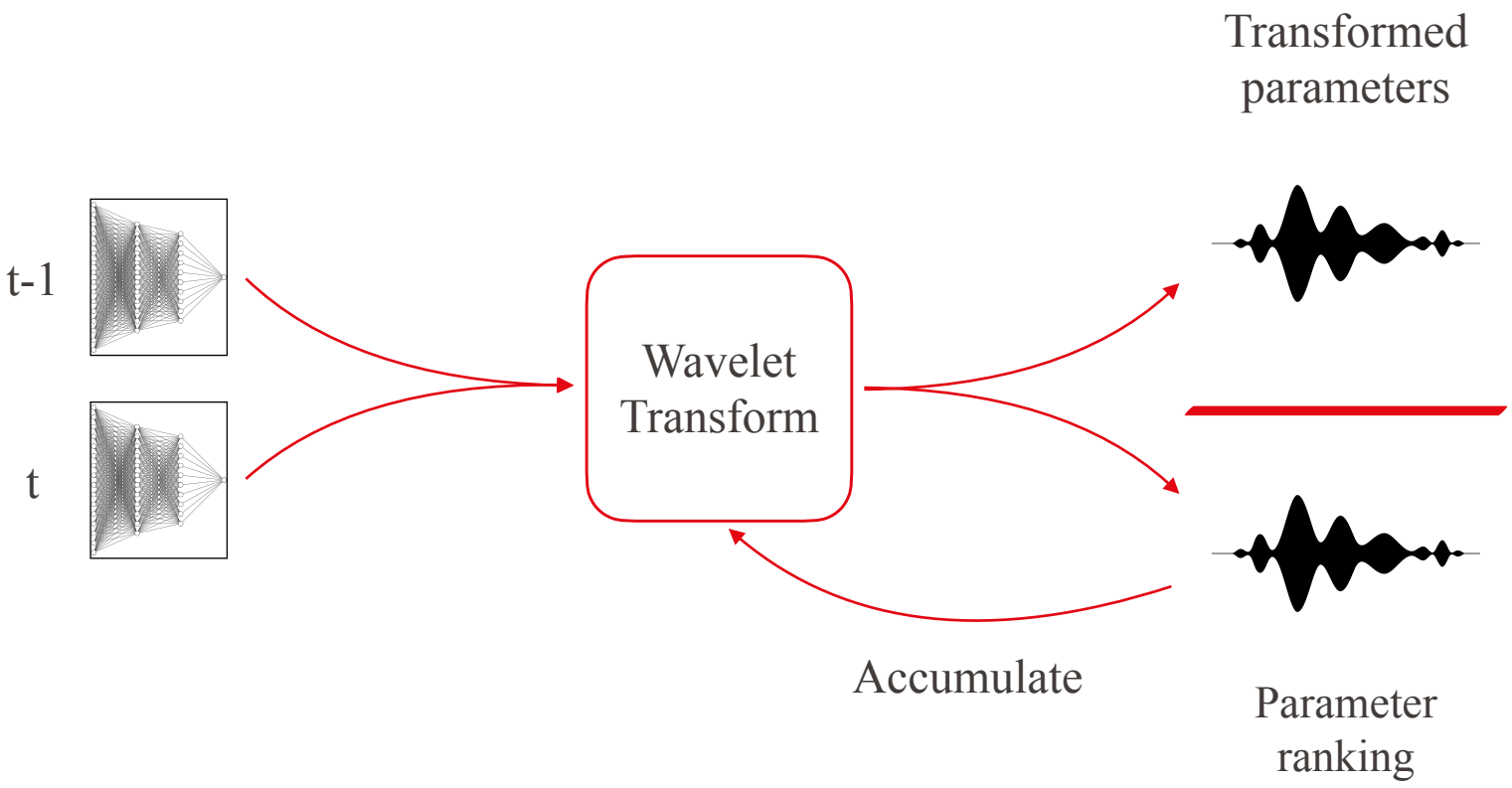
Parameter Ranking



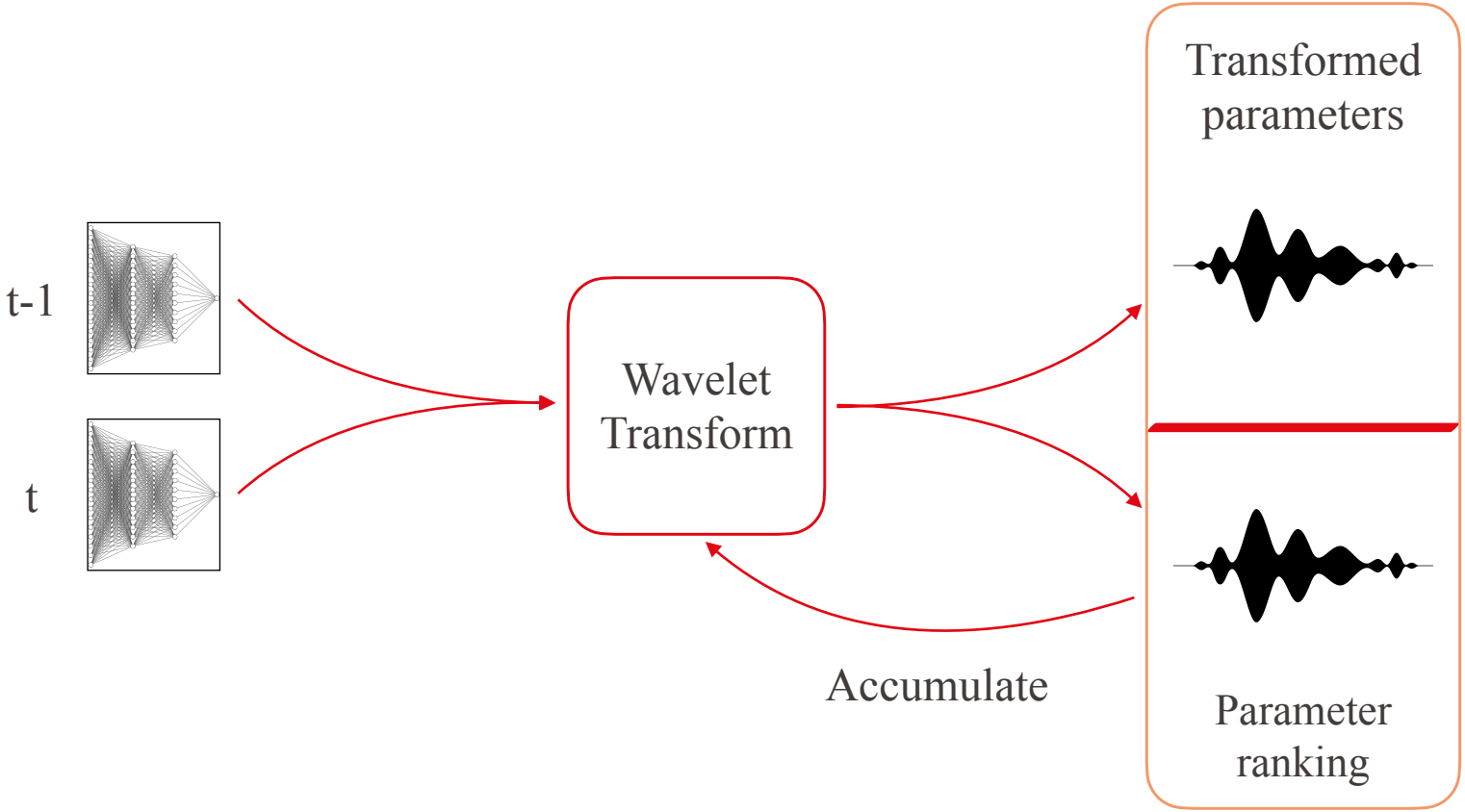
Parameter Ranking



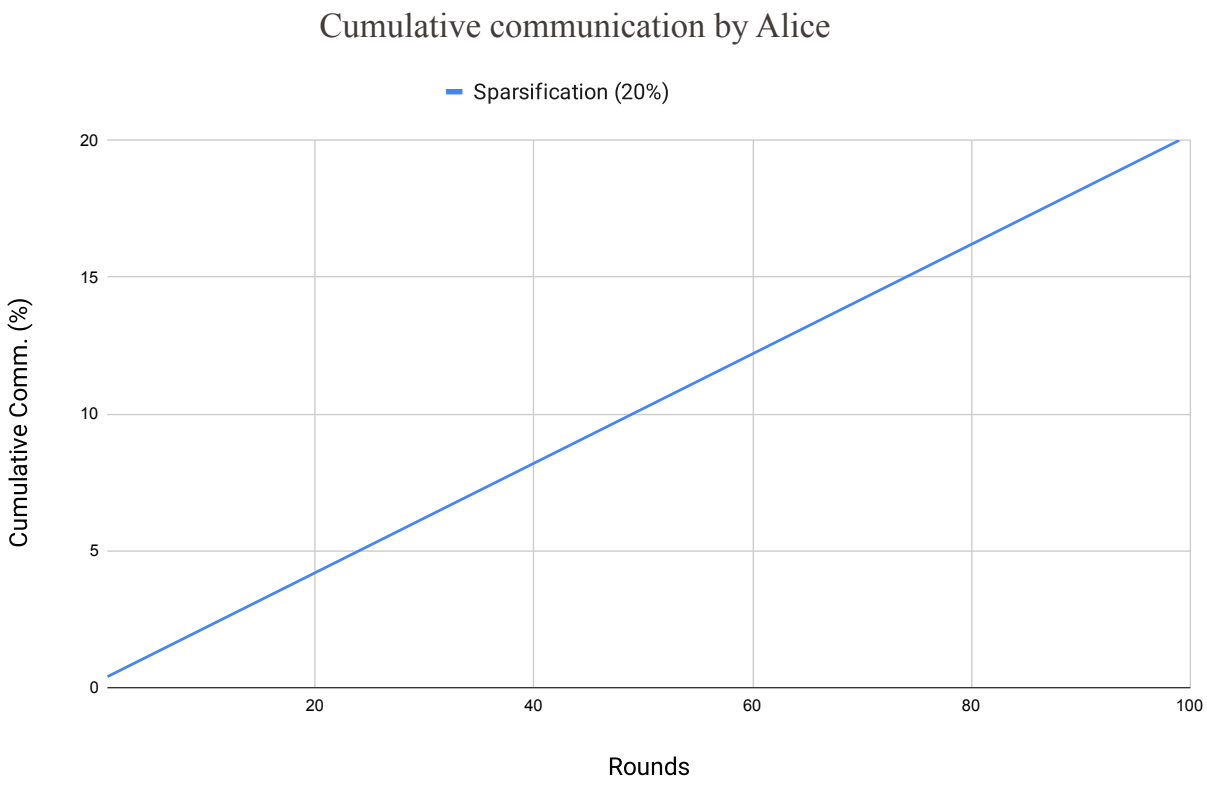
Parameter Ranking



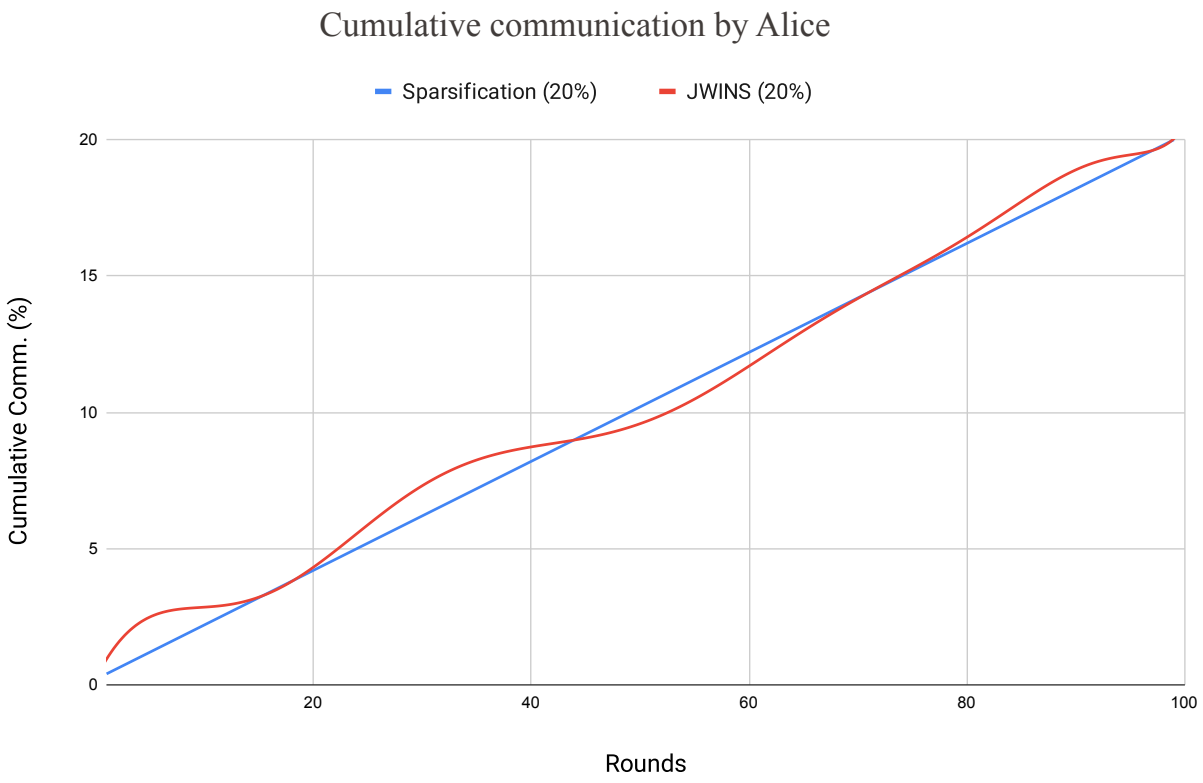
Parameter Ranking



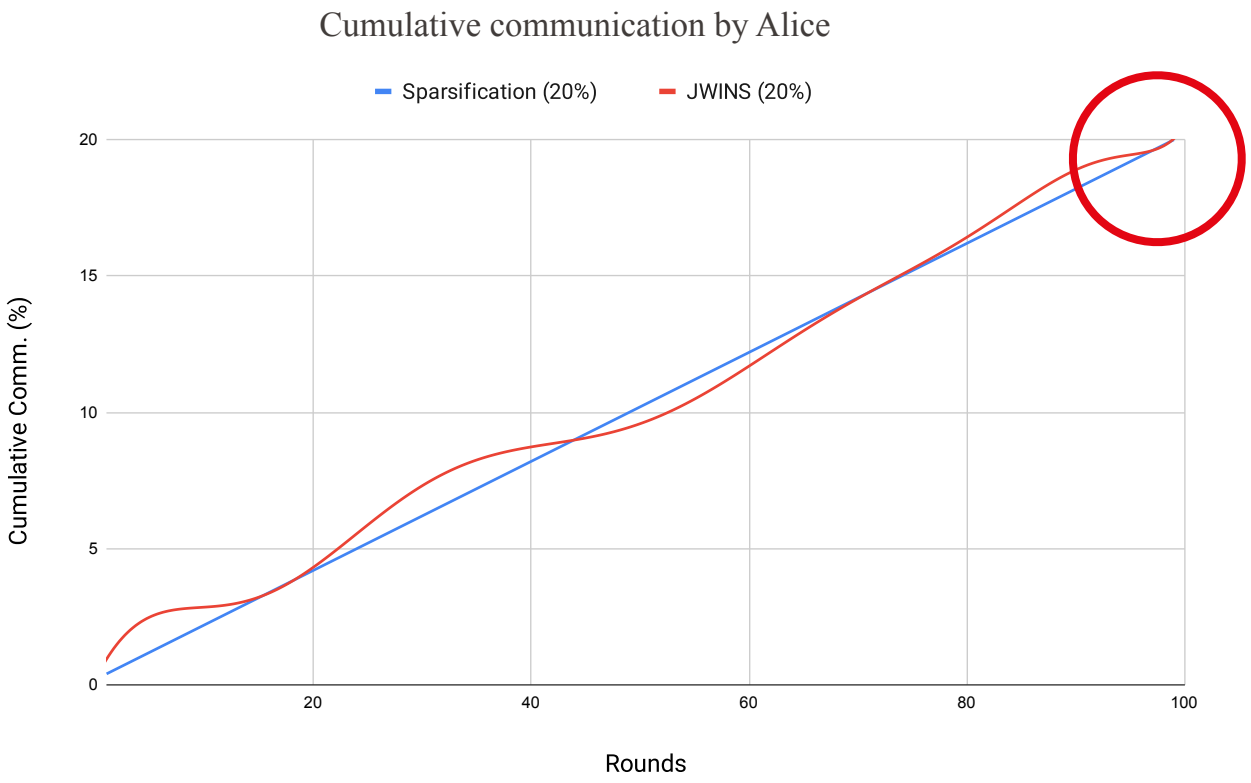
Parameter Selection



Parameter Selection



Parameter Selection



Quick bypass of accumulation.

Overhead Optimization

Original vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Ordered by position

Overhead Optimization

Original vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Ordered by position

Sparsified vector to share

		0.09	0.11	-0.23		0.12				0.98	0.02

Overhead Optimization

Original vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Ordered by position

Sparsified vector to share

0.09	0.11	-0.23	0.12	0.98	0.02

Indices

2	3	4	6	10	11

Overhead Optimization

Original vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

Ordered by position

Sparsified vector to share

0.09	0.11	-0.23	0.12	0.98	0.02

Indices

2	3	4	6	10	11

Size doubled,
Compression required

Overhead Optimization

Original vector

0.65	0.02	0.09	0.11	-0.23	-0.04	0.12	0.34	-0.75	-0.01	0.98	0.02

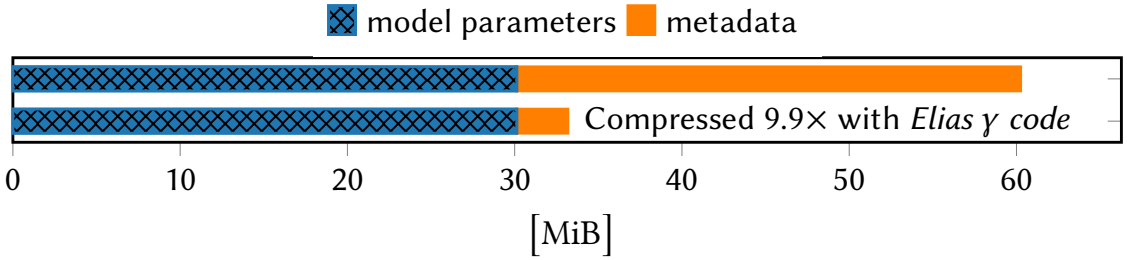
Ordered by position

Sparsified vector to share

Indices

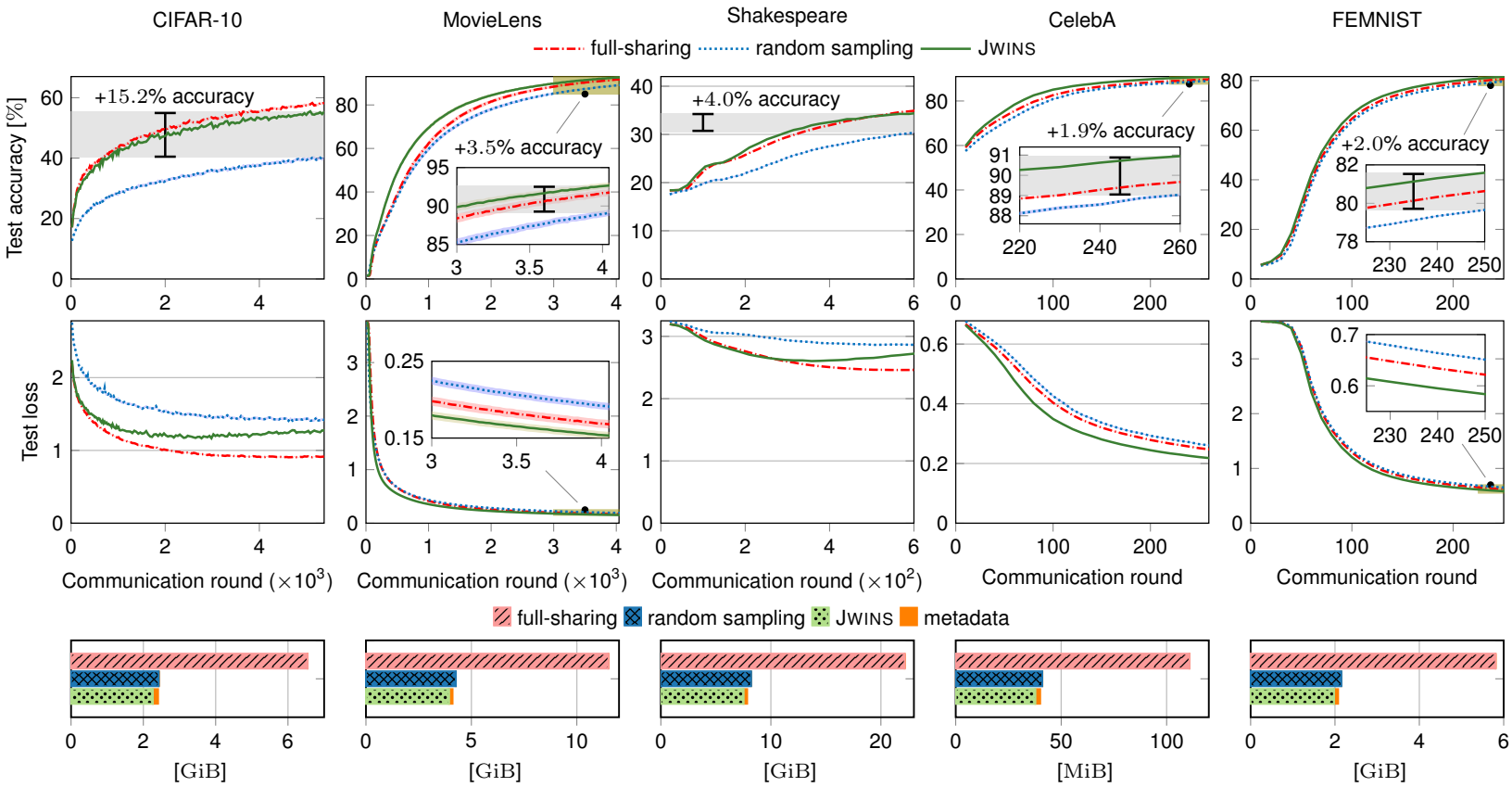
0.09	0.11	-0.23	0.12	0.98	0.02
2	3	4	6	10	11

Size doubled,
Compression required

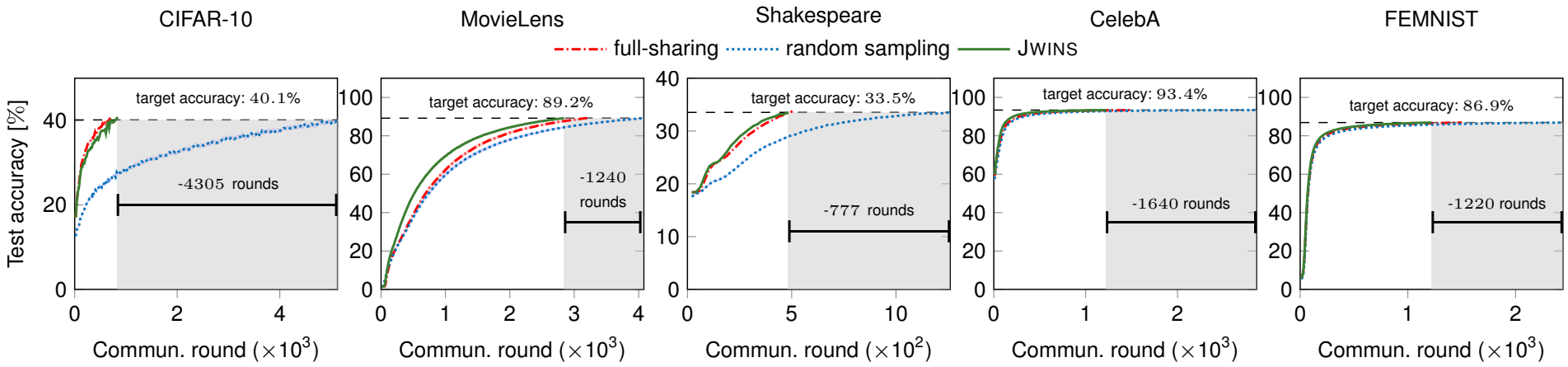


Evaluation: Equal Rounds

96 Nodes, 4-Regular Random Graph
6 machines, 16 procs/machine, niid



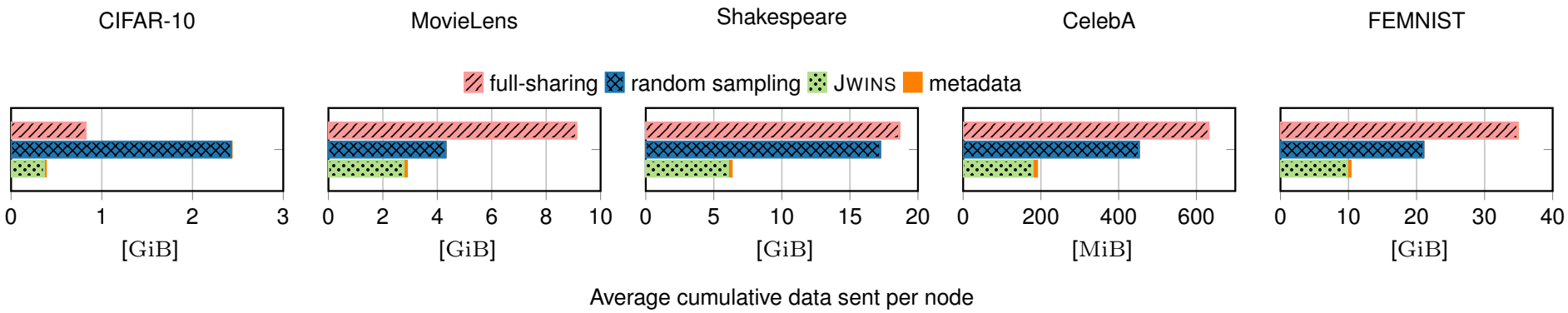
Evaluation: Target Accuracy



Communication budget: 37%

1.5x - 4x communication efficiency!

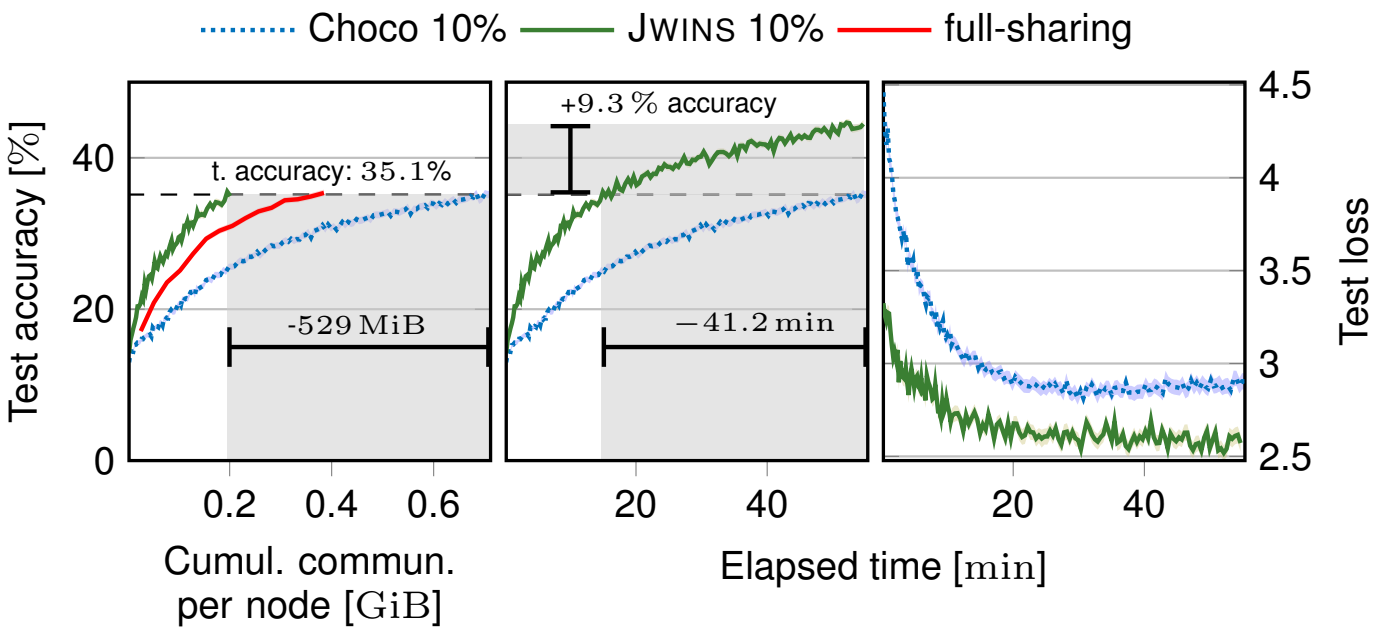
Evaluation: Target Accuracy



Communication budget: 37%

1.5x - 4x communication efficiency!

Evaluation: Choco-SGD

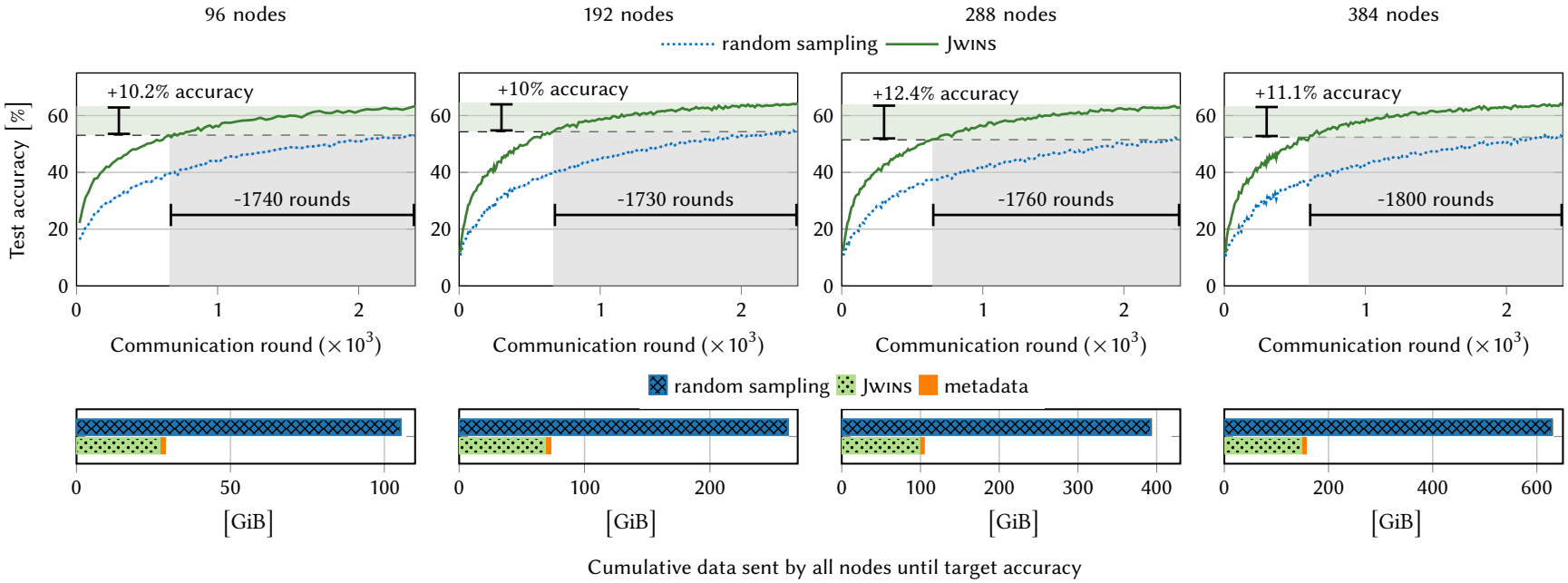


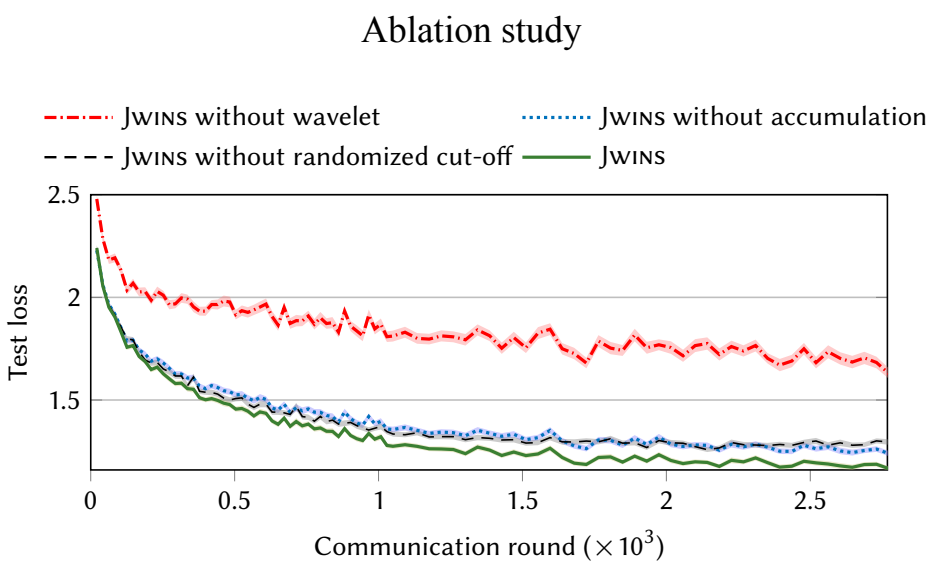
3.9x faster!

Consistent performance at lower budgets.

Evaluation: Scalability

N Nodes, 4-Regular Random Graph
16 machines, N/16 procs/machine, niid



More results

- Scalable beyond 96 nodes.
- Competitive against full-sharing.
- Wavelet transform makes the difference.