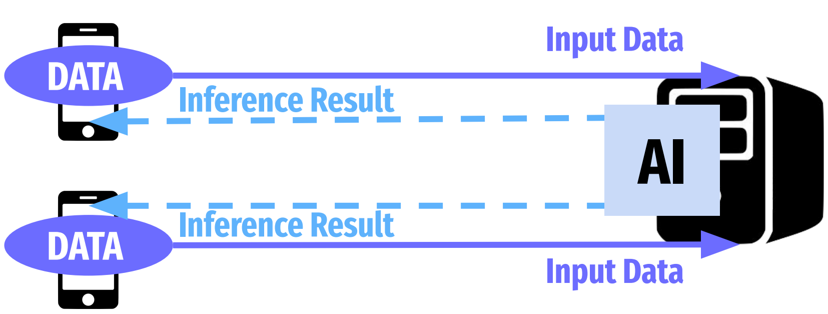
Sparse Communication for Federated Learning

Kundjanasith Thonglek[1], Keichi Takahashi[2], Kohei Ichikawa[1], Chawanat Nakasan[3], Pattara Leelaprute[4], and Hajimu Iida[1]

[1] Nara Institute of Science and Technology, Nara, Japan
[2] Tohoku University, Sendai, Japan
[3] Kanazawa University, Ishikawa, Japan
[4] Kasetsart University, Bangkok, Thailand

Deployment approaches for AI applications

Cloud-based AI



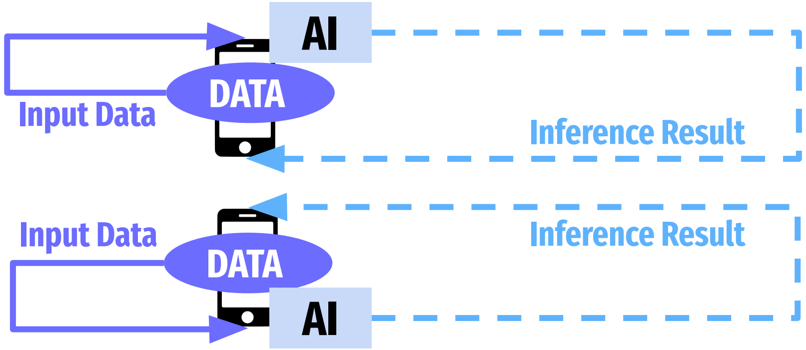
Pros

- ➢ Model is trained using data from all edge devices

Cons

- ➢ Longer response time
- ➢ Poor data privacy

Edge-based AI



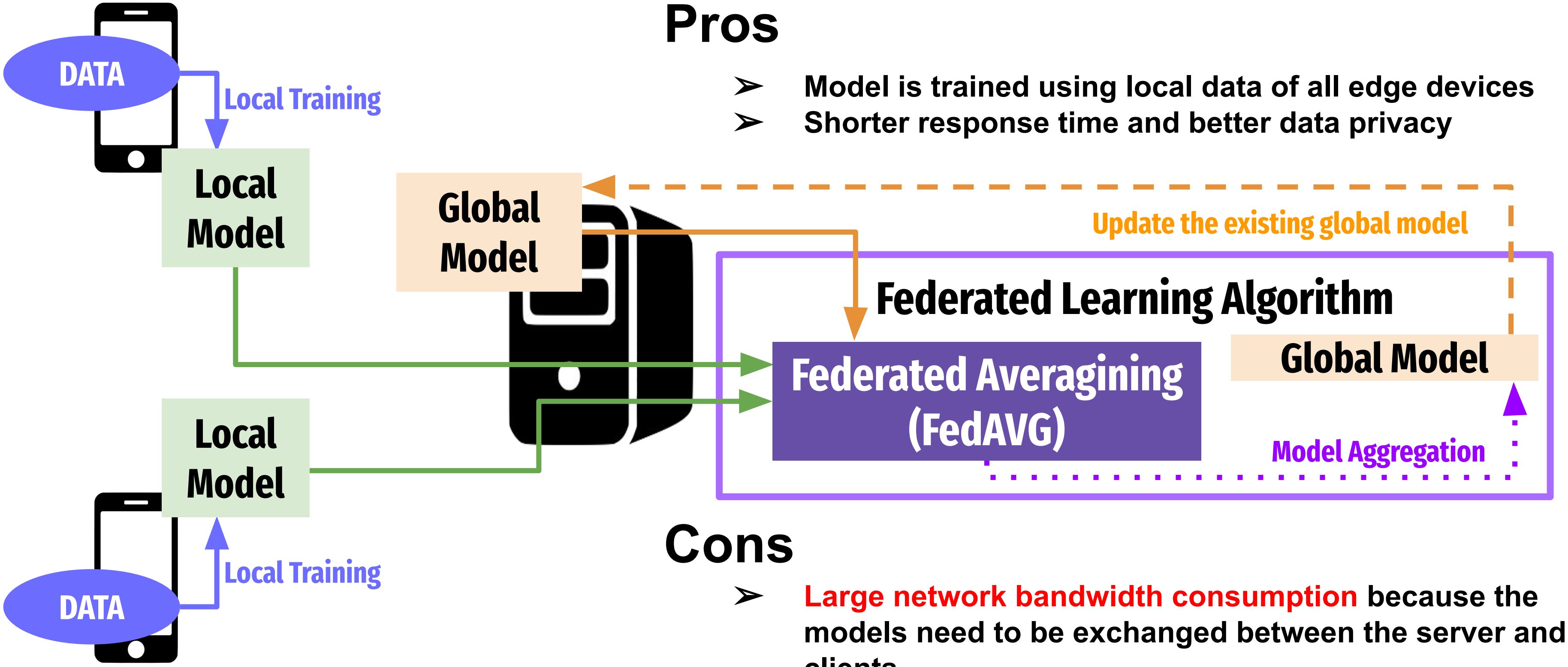
Pros

- ➢ Shorter response time
- ➢ Better data privacy

Cons

- ➢ Edge devices cannot share their data with other devices

Federated learning



Pros

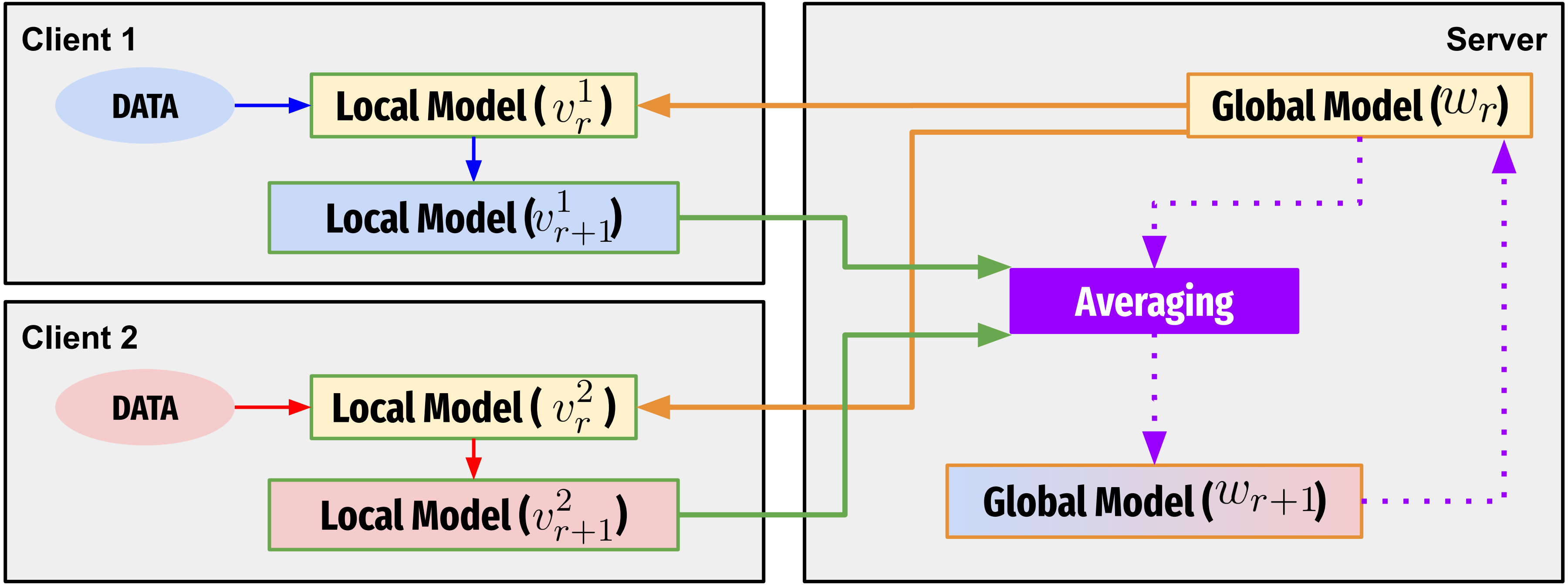
- Model is trained using local data of all edge devices
- Shorter response time and better data privacy

Cons

- Large network bandwidth consumption because the models need to be exchanged between the server and clients

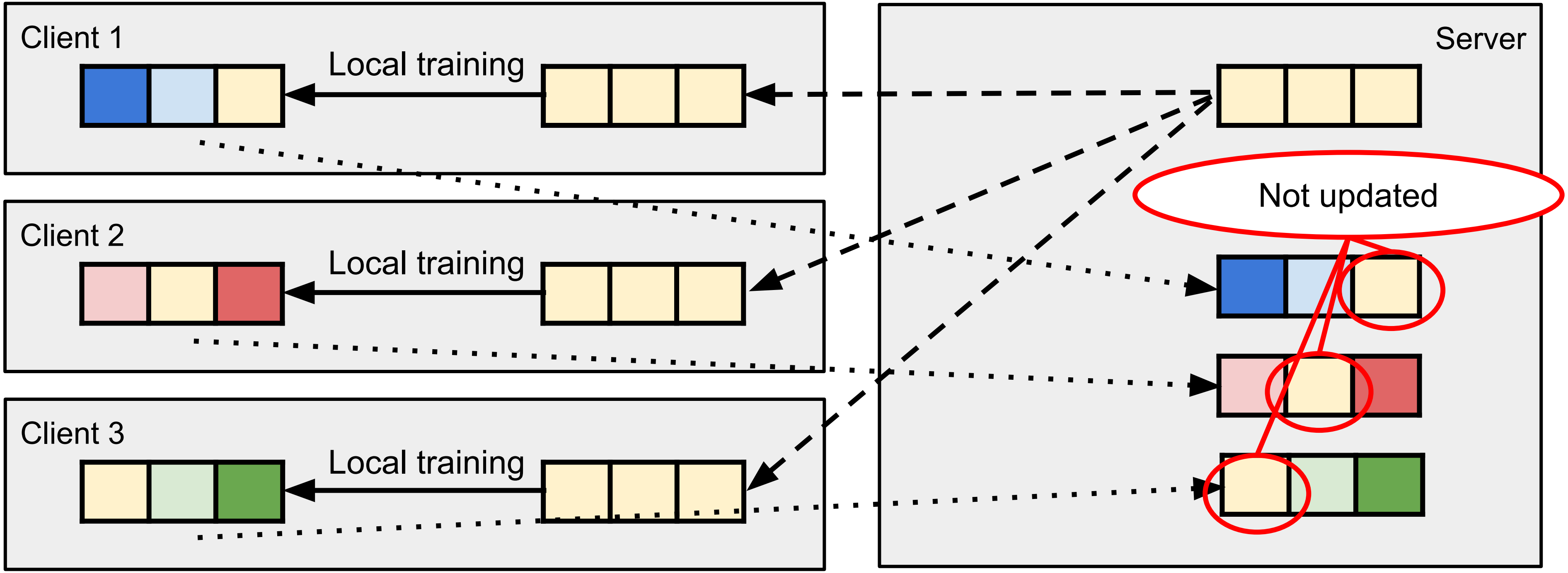
Federated averaging (FedAVG)

w is the weights of the global model on the server
v is the weights of the local model on the client



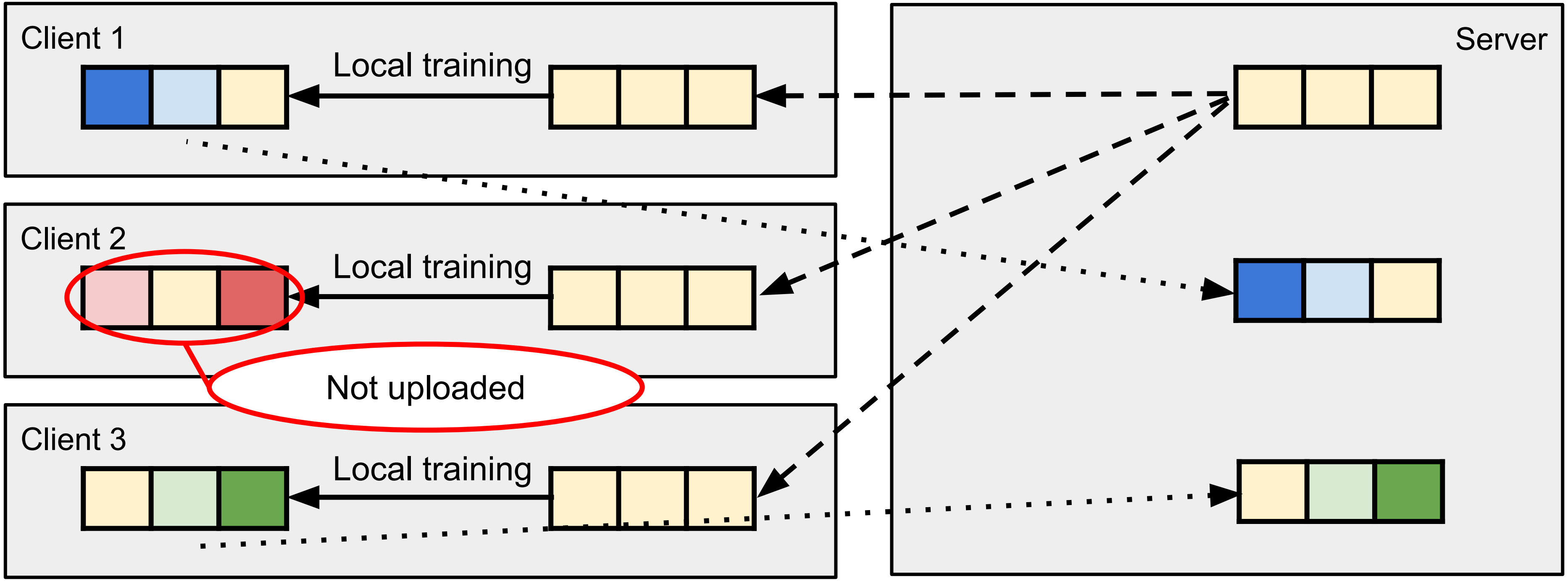
The number of selected clients in each round is $R = C \times N, 0 < C \leq 1,$
where **C** is the fraction of selected clients and **N** is the total number of clients.

Downside #1: Whole models are exchanged



Since the whole models are exchanged between the server and clients, **transferring unupdated parameters** wastes the network bandwidth.

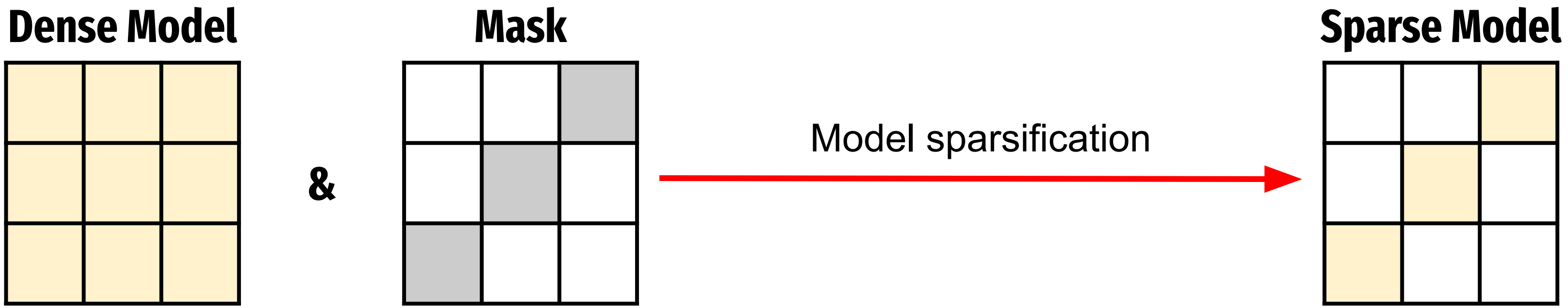
Downside #2: Only a subset of clients participate



Since only a subset of clients participate in one round, **the server misses local updates** that could have been obtained from the excluded clients.

Model sparsification

Model sparsification **omits some parameters** in the dense model to build a sparse model while keeping the same model architecture

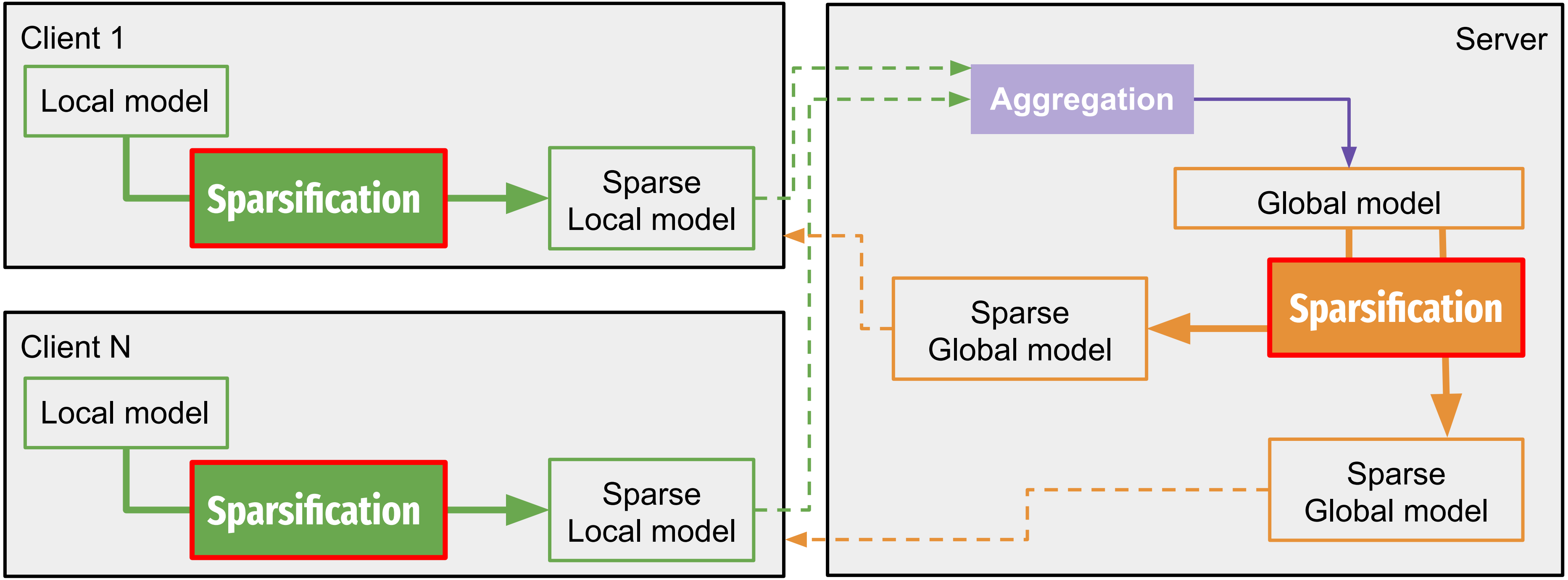


The proposed method exchanges **the most updated parameters** of model between server and clients

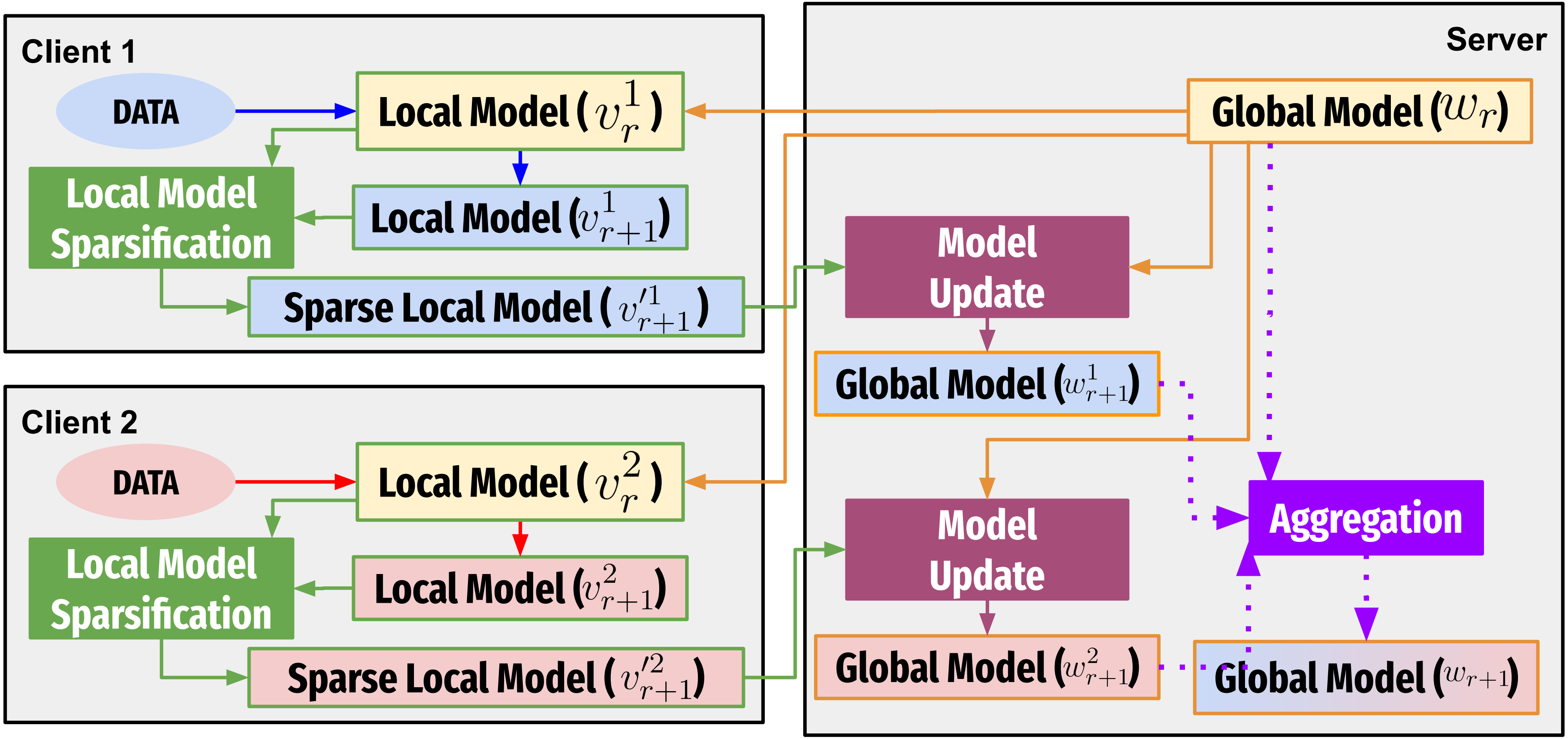
Parameters that are significantly changed after training are expected to have **large impact on the model performance**

Basic idea behind the proposed method

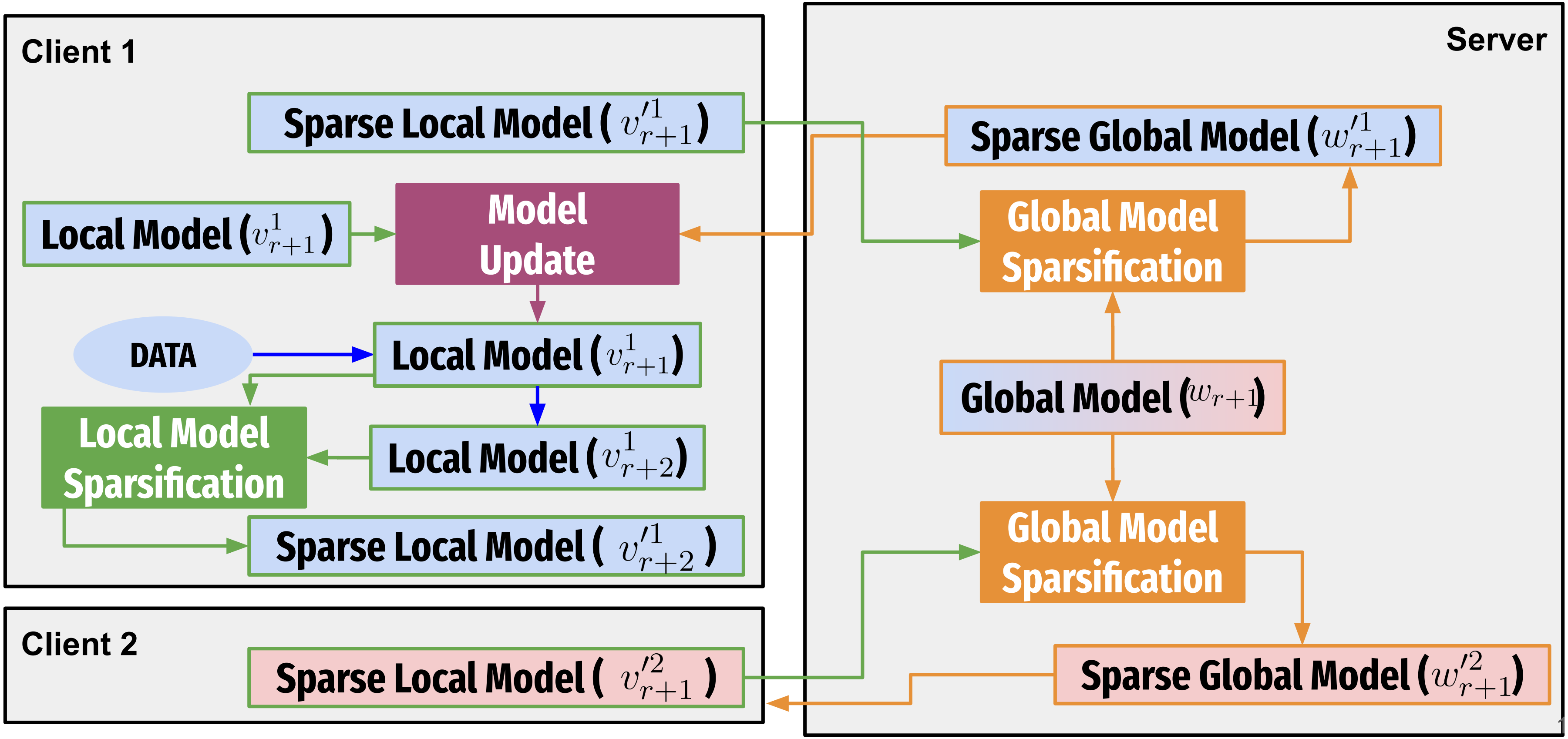
Sparsify the models exchanged between the server and clients in both directions



Overview of the proposed method (Uplink)

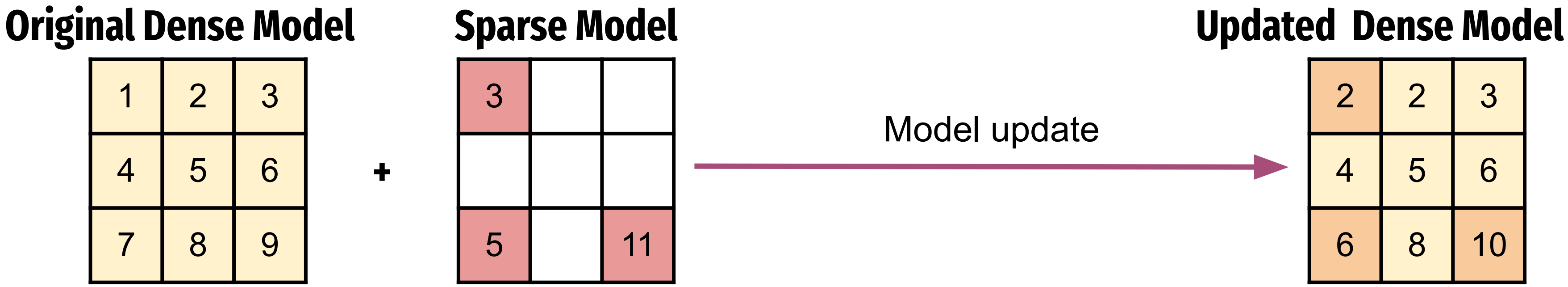


Overview of the proposed method (Downlink)



Model update

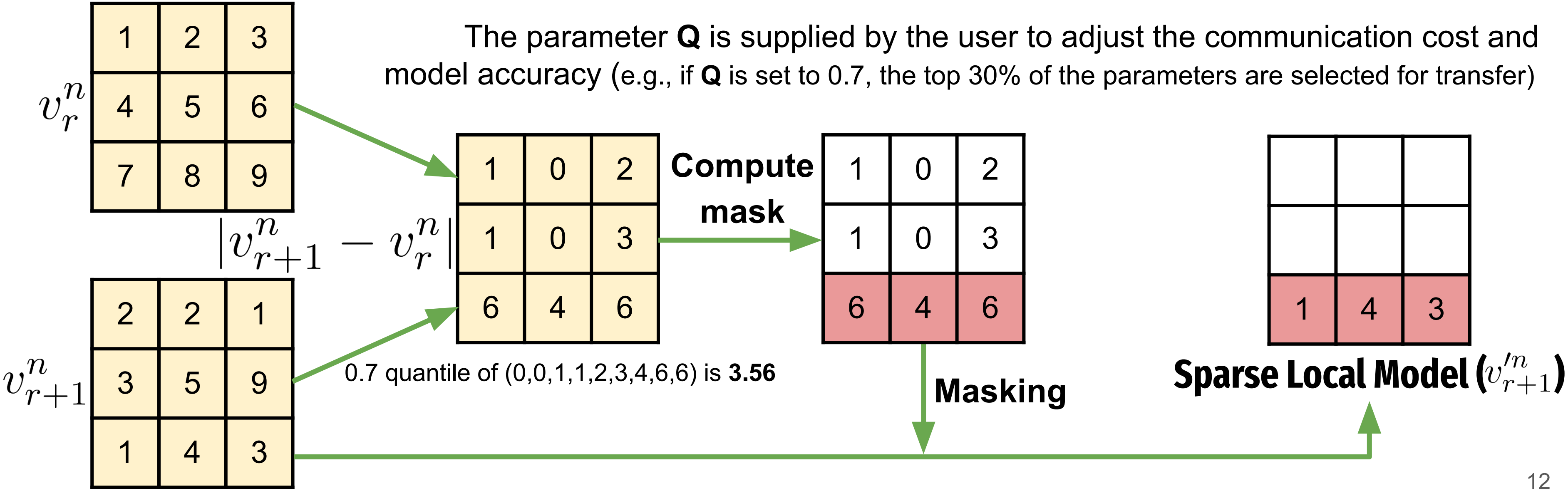
Updates a dense model using sparse updates sent from the server or clients



Local model sparsification

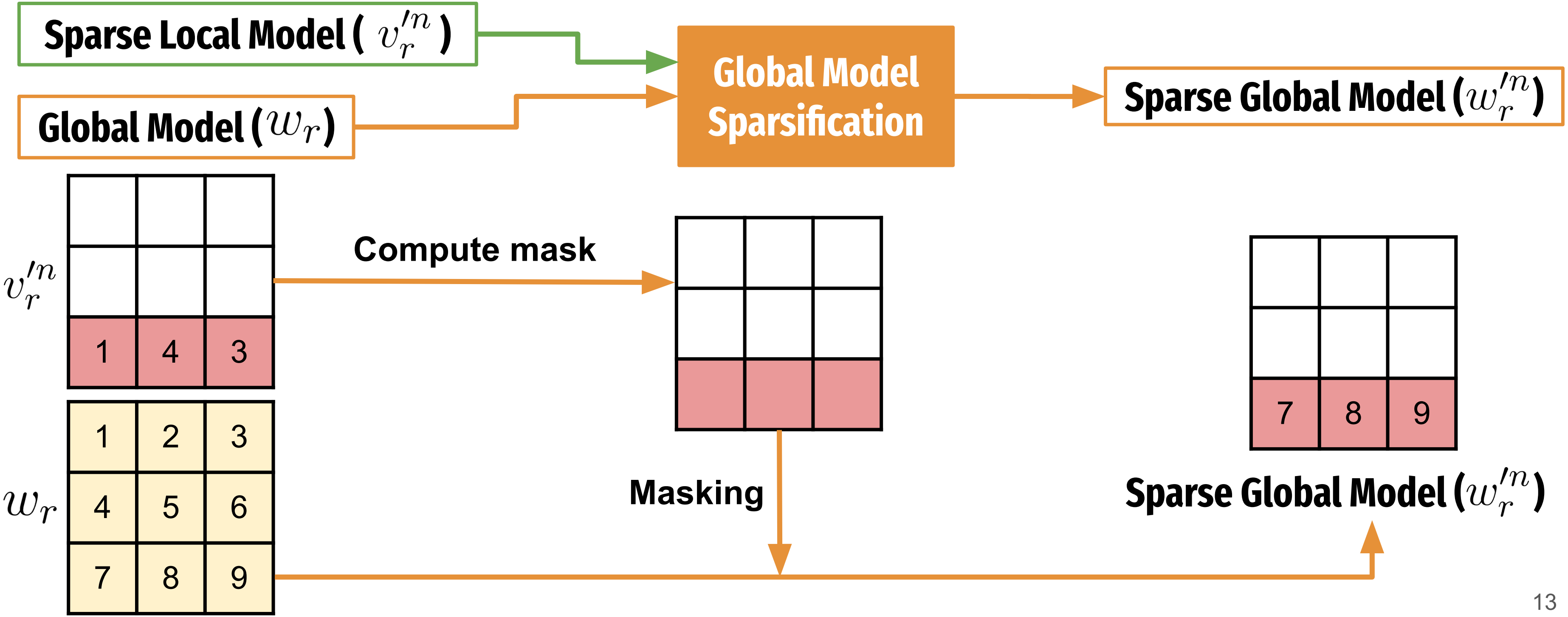
Extracts the most updated parameters to construct sparse local model

The parameter **Q** is supplied by the user to adjust the communication cost and model accuracy (e.g., if **Q** is set to 0.7, the top 30% of the parameters are selected for transfer)



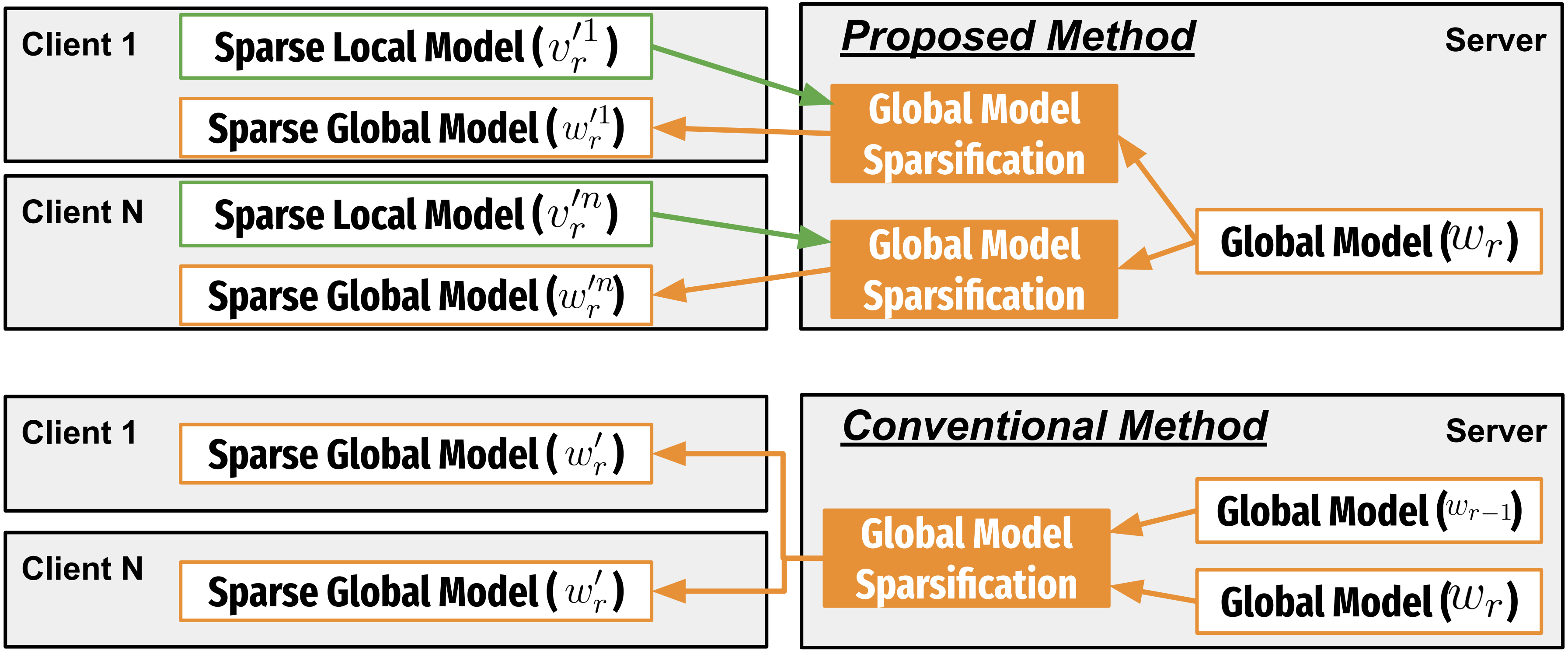
Global model sparsification

Reuse local model mask to construct sparse global model because the parameters in the mask are still not converged yet at client-side and then those parameters have to be updated to converge



Proposed method vs Conventional method

Downlink communication is the main difference between the proposed and conventional methods



Experimental environment

➢ Models:

1. VGG16 (553.43 MB)
2. ResNet152 (243.21 MB)
3. DenseNet201 (89.92 MB)
4. MobileNet (17.02 MB)

➢ Datasets:

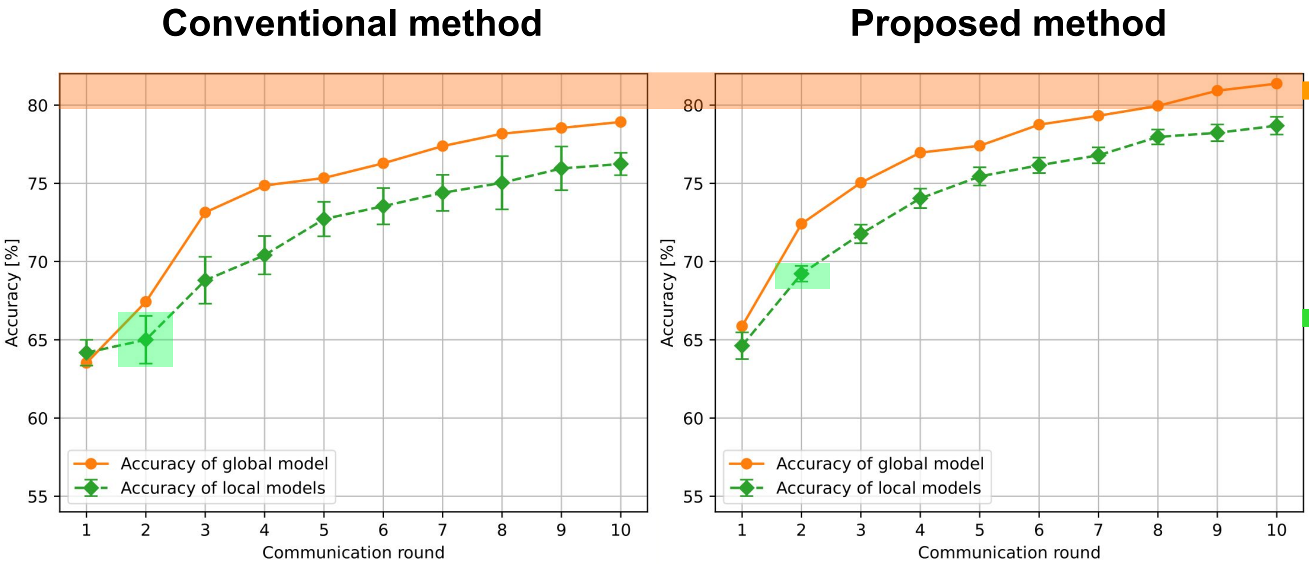
1. CIFAR-10
2. CIFAR-100
3. MNIST
4. FMNIST

Experimental Setup

Configuration	Value
# of communication rounds *(R)*	10
# of clients *(N)*	10
# of local epochs *(E)*	5
Local batch size *(B)*	8

Although large models can generally achieve higher accuracy than small models, not all edge devices can deploy large models due to **resource constraints**. Thus, we evaluated the proposed method using models with different scales

Comparison to conventional method



The **global model** accuracy of the proposed method is **higher** than that of the conventional method

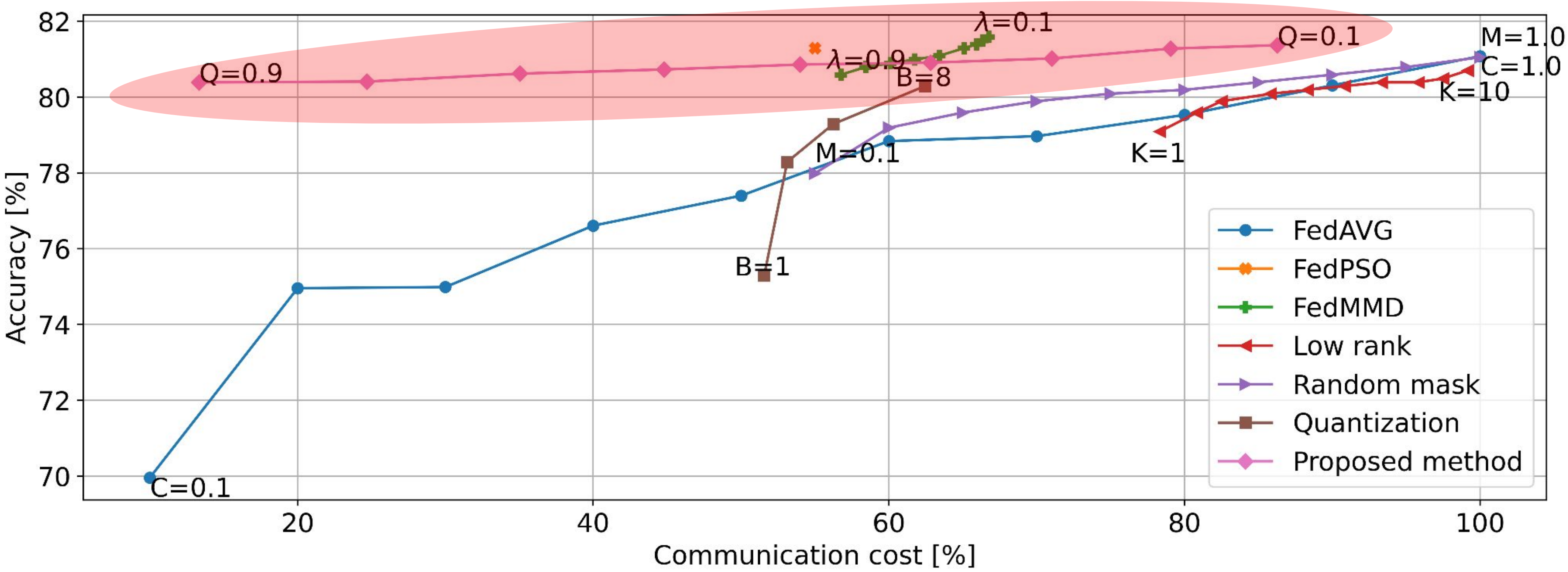
The variance of **local model** accuracy in the proposed method is **lower** than that of the conventional method

Since the conventional method sends **the same global model to all clients**, it is unable to build highly accurate local models, which also leads to a decrease in the accuracy of the global model

Existing methods and their hyperparameters

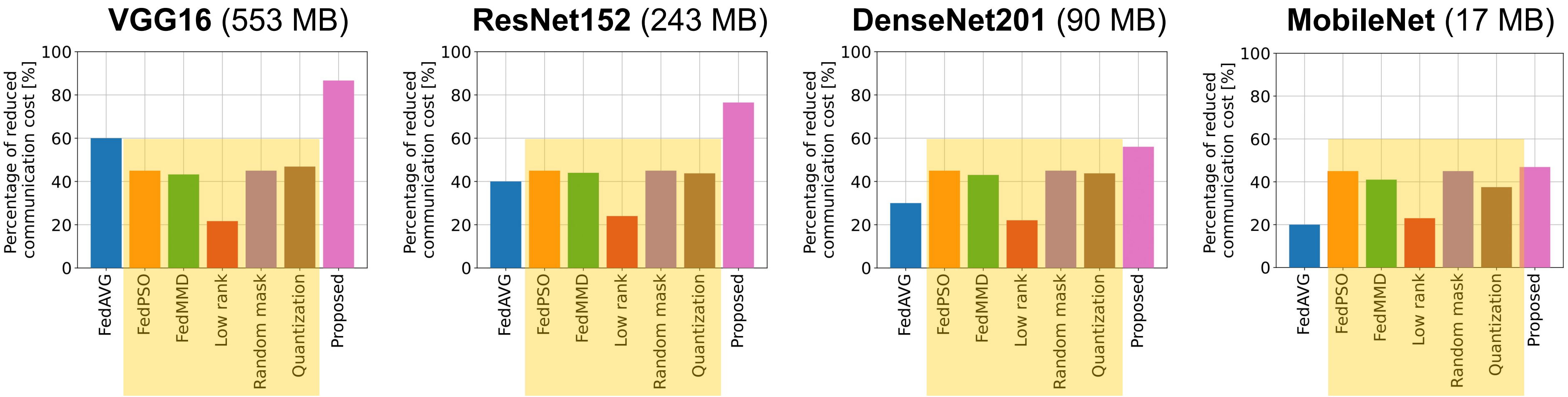
Method name	Hyperparameter	Description
FedAVG	C	Fraction of clients selected in each communication round
FedPSO	N/A	Does not have a hyperparameter to control communication cost
FedMMD	λ	Coefficient of MMD loss between the global and local models
Low rank approximation	K	Rank of the low-rank matrix to be converted
Random mask	M	Size of random mask to generate a random pattern
Quantization	B	Quantized bits used for bit-quantization

Comparison to the existing methods



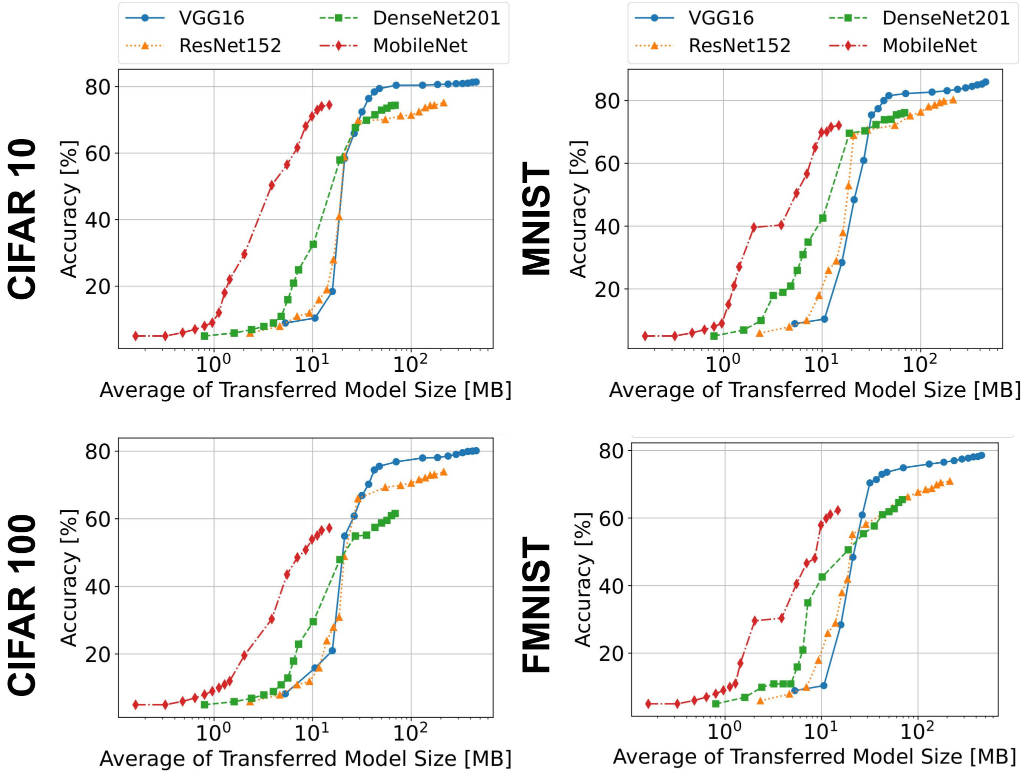
The proposed method outperforms the existing methods in terms of both the communication cost and the accuracy of the global model

Results for different models



- ➢ The reduction of the communication cost from FedPSO, FedMMD, Low rank approximation, Random mask, and Quantization **are almost identical for all model architecture**
- ➢ The reductions of communication cost in FedAVG and Proposed method **depend on the size of each model architecture** (larger models are more compressed than smaller models)

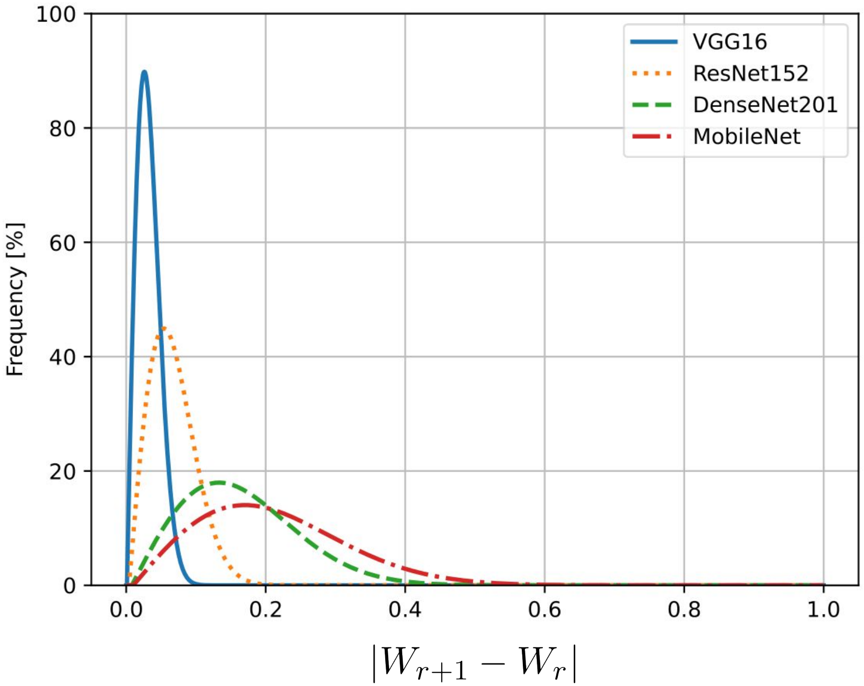
Results for different datasets



- ➢ The proposed method is evaluated over four image classification datasets
- ➢ Q is varied from 0.1 to 0.9 at intervals of 0.1, and from 0.91 to 0.99 at intervals of 0.01
- ➢ The proposed method produces consistent results for all datasets
  - ➢ **It is able to reduce the amount of data transfer for larger models than for smaller models without a significant loss of accuracy**

Why are larger models amenable to compression?

Frequency of updated values
per communication round on a client



- ➢ In larger models (e.g., VGG16), small parameter updates are more frequent than in smaller models (e.g., MobileNet)
  - ○ Small parameter updates have a smaller impact on the model performance
- ➢ Large models receive more low-impact updates than small models
  - ○ **The proposed method drops those low-impact updates in large models without a significant loss of accuracy**

Conclusion

- **We proposed a novel method to reduce the communication cost for federated learning by sparsifying local and global models on both uplink and downlink communication**

- **The proposed method utilizes exchanging the most updated parameters of neural network models**

- **Diverse models and datasets are used to evaluate the proposed method in terms of model accuracy and communication cost**
    - The proposed method achieved a reduction in the communication costs approximately **90%**

Future work

- Architecture of other neural network models should be investigated to improve reducing the required communication cost

- Updating the parameters in other neural network models should be observed during the local training procedure

- Large number of edge devices should be used to evaluate the efficiency of the proposed method

Q & A

Thank you for your attention

Email: thonglek.kundjanasith.ti7@is.naist.jp