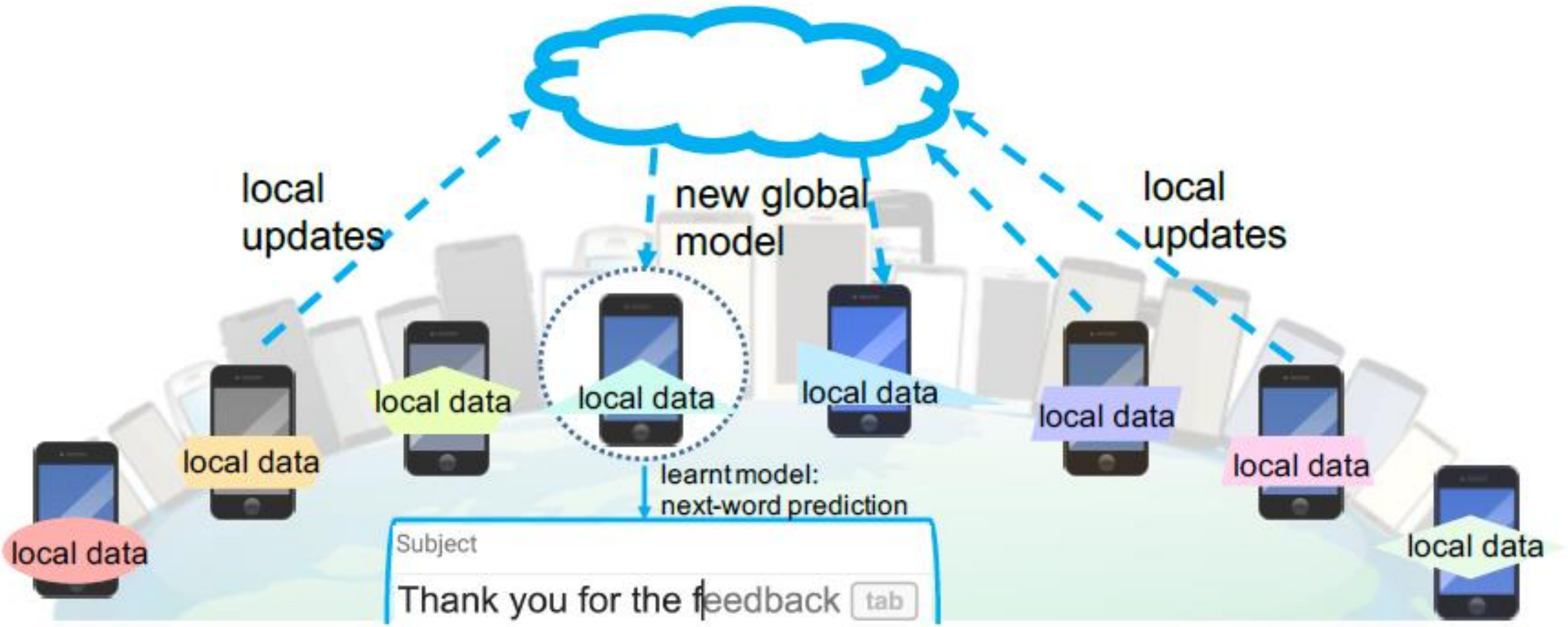
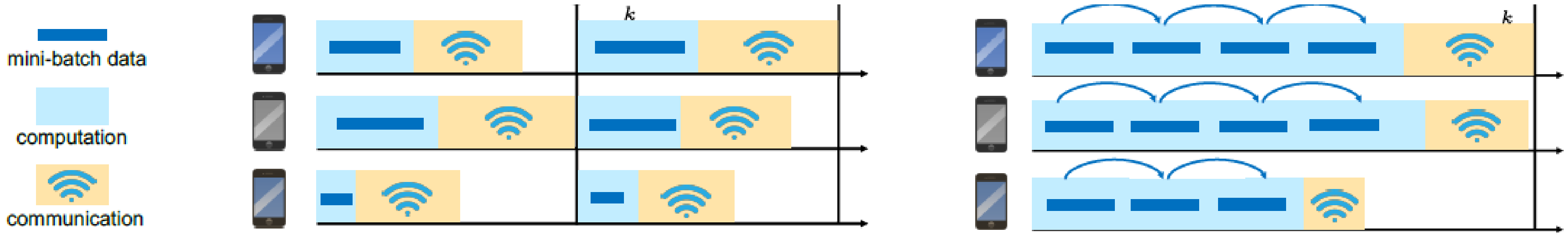
FEDERATED LEARNING

곽소진

Federated Learning



Local Updating

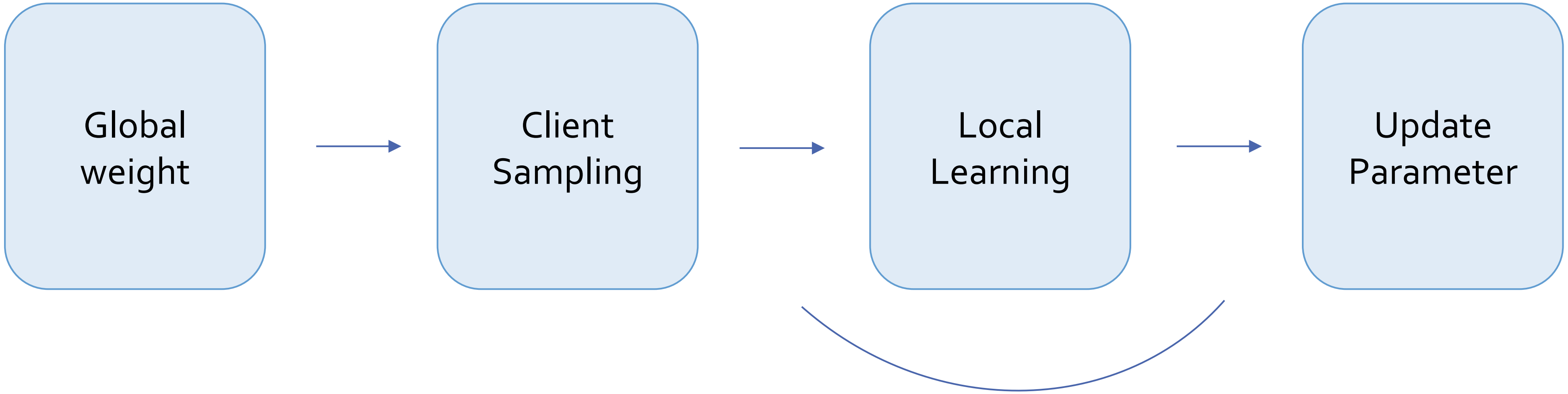


FedAvg

- FedSGD : Send every update
- FedAVG : FedSGD + minibatch

  : repeatedly update parameter → main server

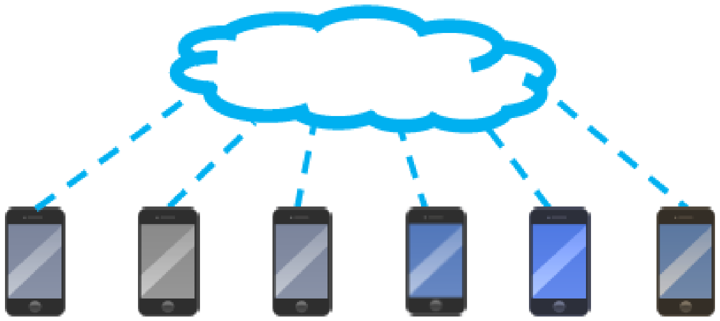
  : least communication cost



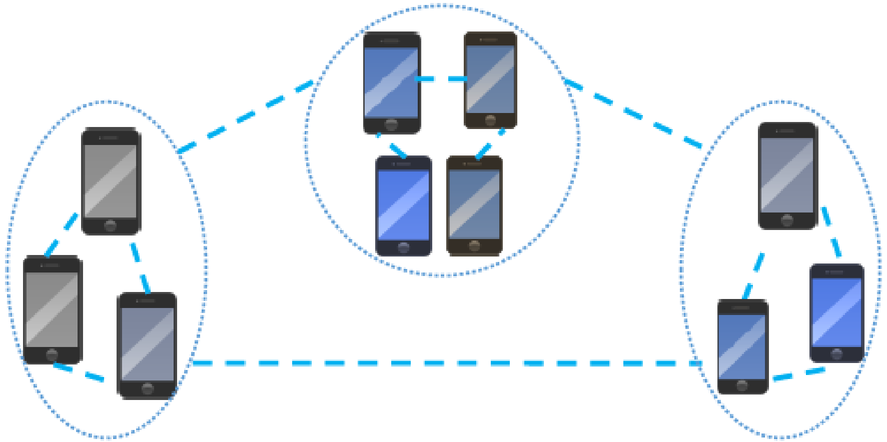
Compression Schemes

Sparsification	Quantization
reducing the amount of data to communicate by zeroing some of the parameters of the model	reducing the data size by expressing the parameter value of the model with a smaller number of bits

Decentralization

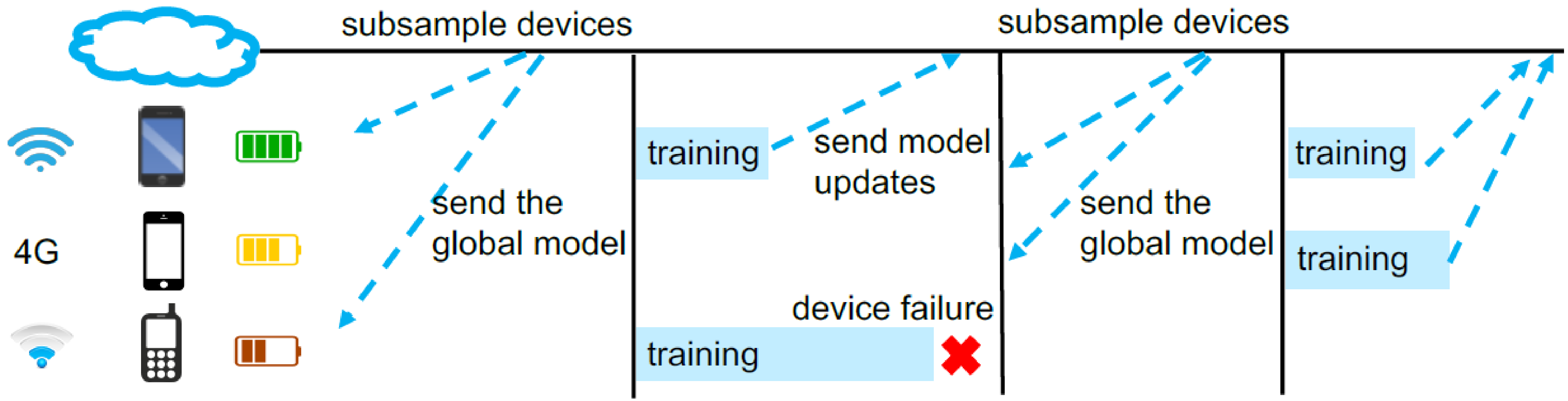


Star



Distributed

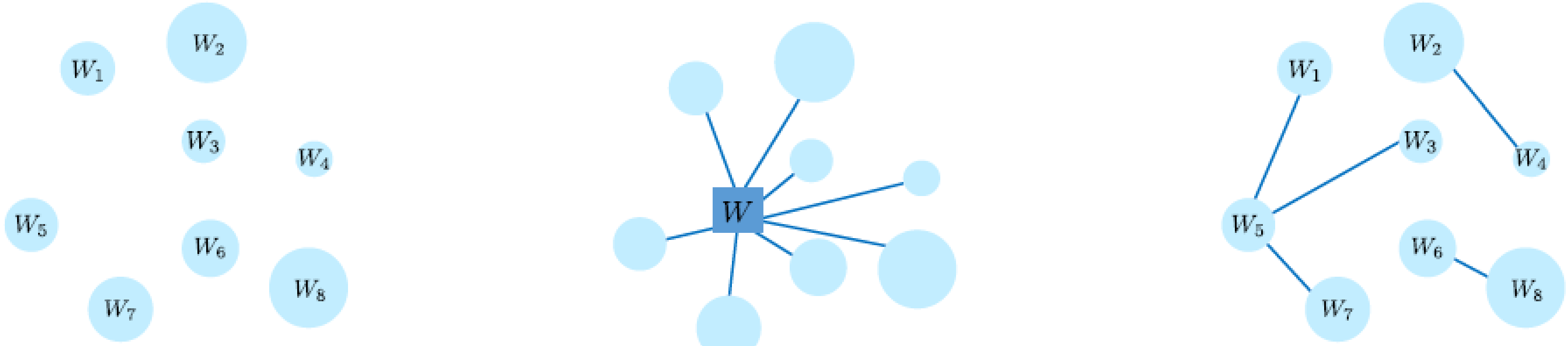
Systems heterogeneity



System Heterogeneity

- Asynchronous Communication
- Active Sampling
- Fault Tolerance : Coded Computation → privacy, network

Statistical Heterogeneity



(a) Learn personalized models for each device; do not learn from peers.

(b) Learn a global model; learn from peers.

(c) Learn personalized models for each device; learn from peers.

Modeling heterogeneous data

- local updates based on local loss
- Agnostic Federated Learning
- q-FFL

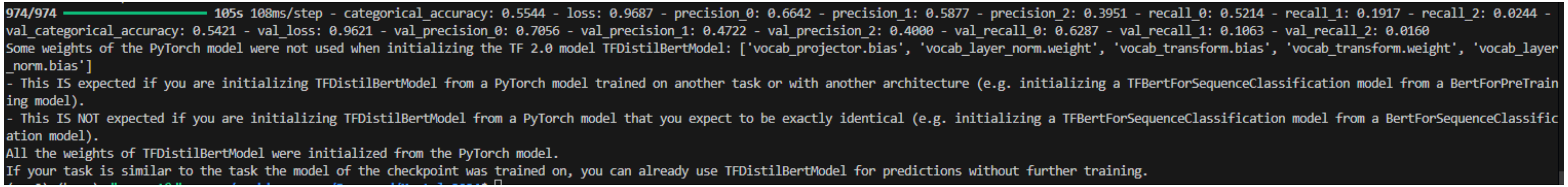
Non -IID

- FedProx
- Better FedAvg
- Provides convergence assurances and helps address heterogeneity between devices
- Privacy & Bandwidth

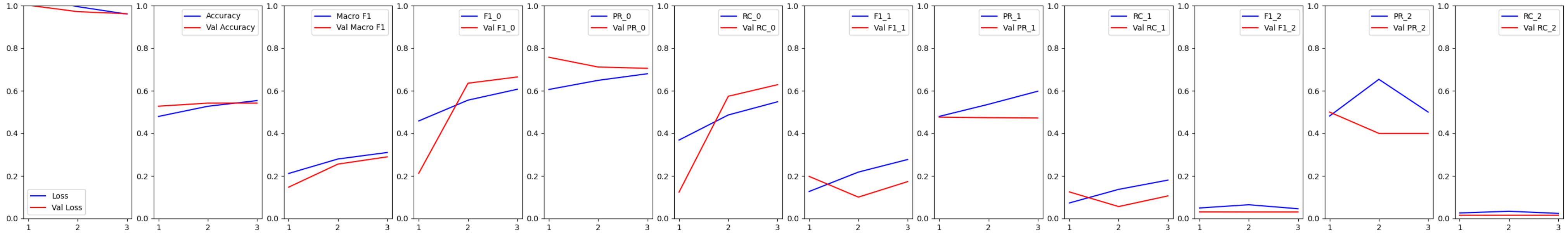
Privacy

- Global & Local
- computationally cheap, communication-efficient
- SMC, differential privacy → various method
- Compression + differential privacy == good

Thank you



```
974/974 ━━━━━━━━━━━━━━━━━━━━ 105s 108ms/step - categorical_accuracy: 0.5544 - loss: 0.9687 - precision_0: 0.6642 - precision_1: 0.5877 - precision_2: 0.3951 - recall_0: 0.5214 - recall_1: 0.1917 - recall_2: 0.0244 -
val_categorical_accuracy: 0.5421 - val_loss: 0.9621 - val_precision_0: 0.7056 - val_precision_1: 0.4722 - val_precision_2: 0.4000 - val_recall_0: 0.6287 - val_recall_1: 0.1063 - val_recall_2: 0.0160
Some weights of the PyTorch model were not used when initializing the TF 2.0 model TFDistilBertModel: ['vocab_projector.bias', 'vocab_layer_norm.weight', 'vocab_transform.bias', 'vocab_transform.weight', 'vocab_layer
_norm.bias']
- This IS expected if you are initializing TFDistilBertModel from a PyTorch model trained on another task or with another architecture (e.g. initializing a TFBertForSequenceClassification model from a BertForPreTrain
ing model).
- This IS NOT expected if you are initializing TFDistilBertModel from a PyTorch model that you expect to be exactly identical (e.g. initializing a TFBertForSequenceClassification model from a BertForSequenceClassific
ation model).
All the weights of TFDistilBertModel were initialized from the PyTorch model.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFDistilBertModel for predictions without further training.
```


Loss
Val Loss
Accuracy
Val Accuracy
Macro F1
Val Macro F1
F1_0
Val F1_0
PR_0
Val PR_0
RC_0
Val RC_0
F1_1
Val F1_1
PR_1
Val PR_1
RC_1
Val RC_1
F1_2
Val F1_2
PR_2
Val PR_2
RC_2
Val RC_2