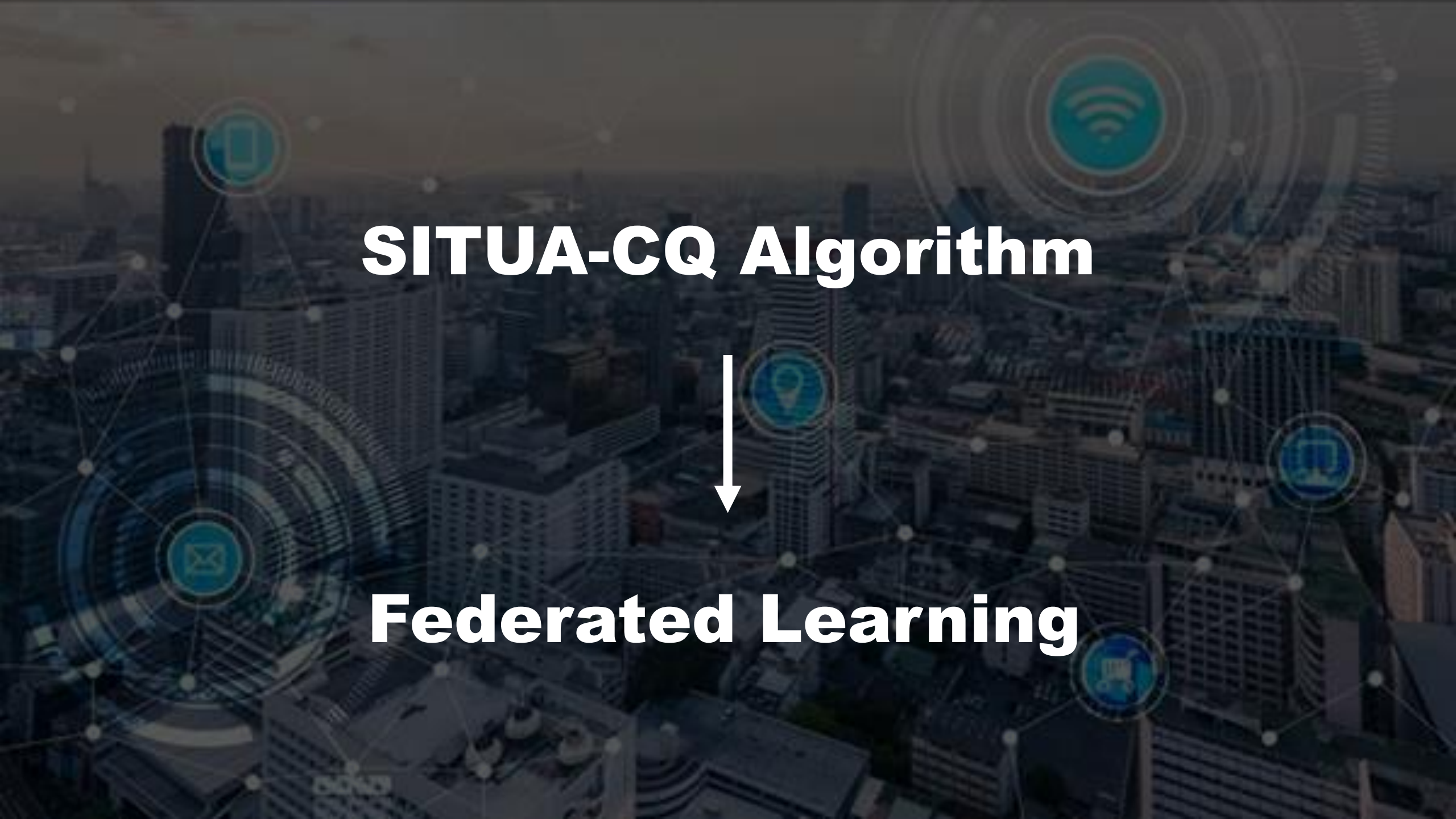
Situation-Aware Cluster and Quantization Level Selection Algorithm for Fast Federated Learning


SITUA-CQ Algorithm
Federated Learning

Abstract

I. Federated Learning

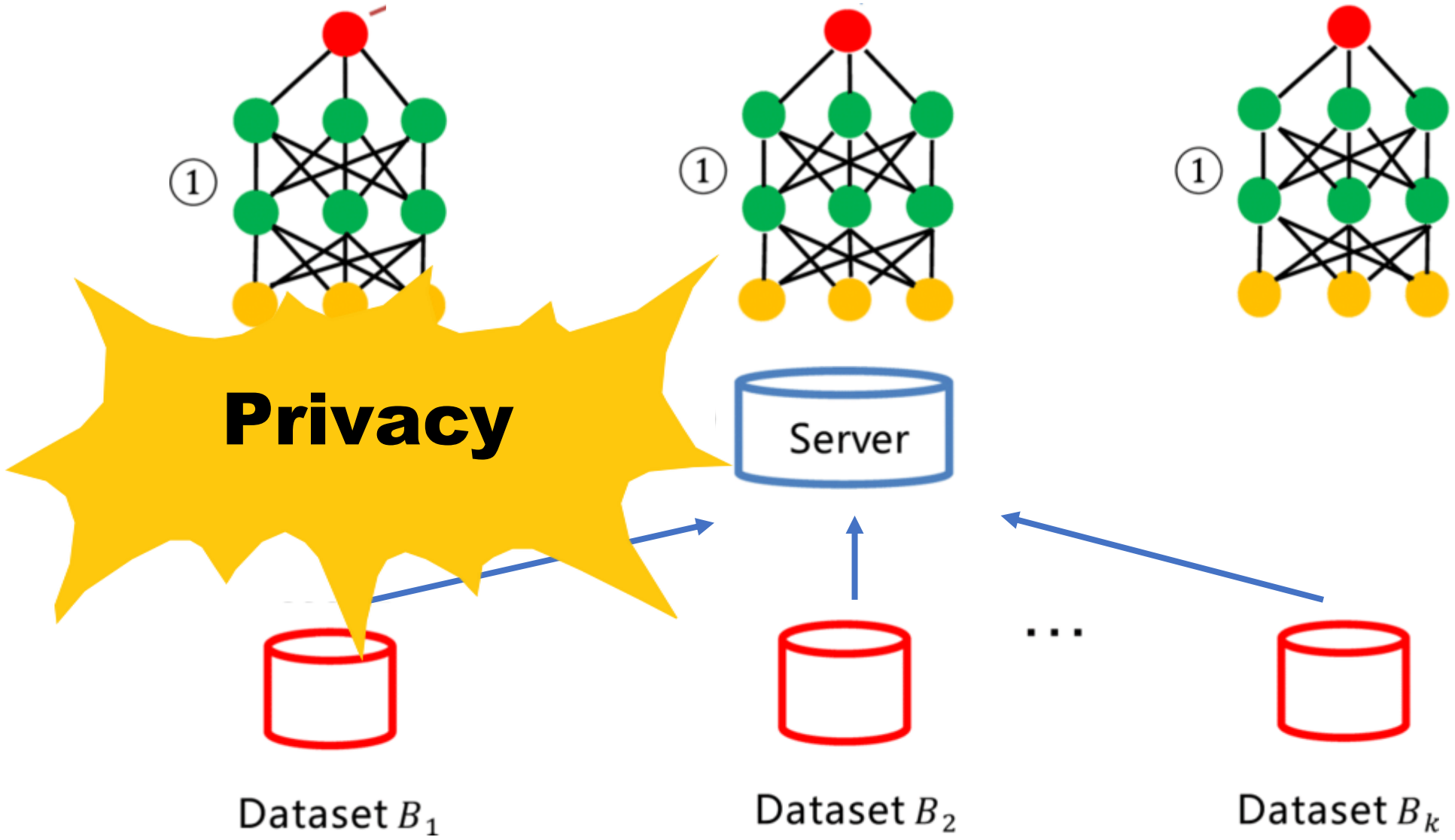
II. System model

III. Accuracy

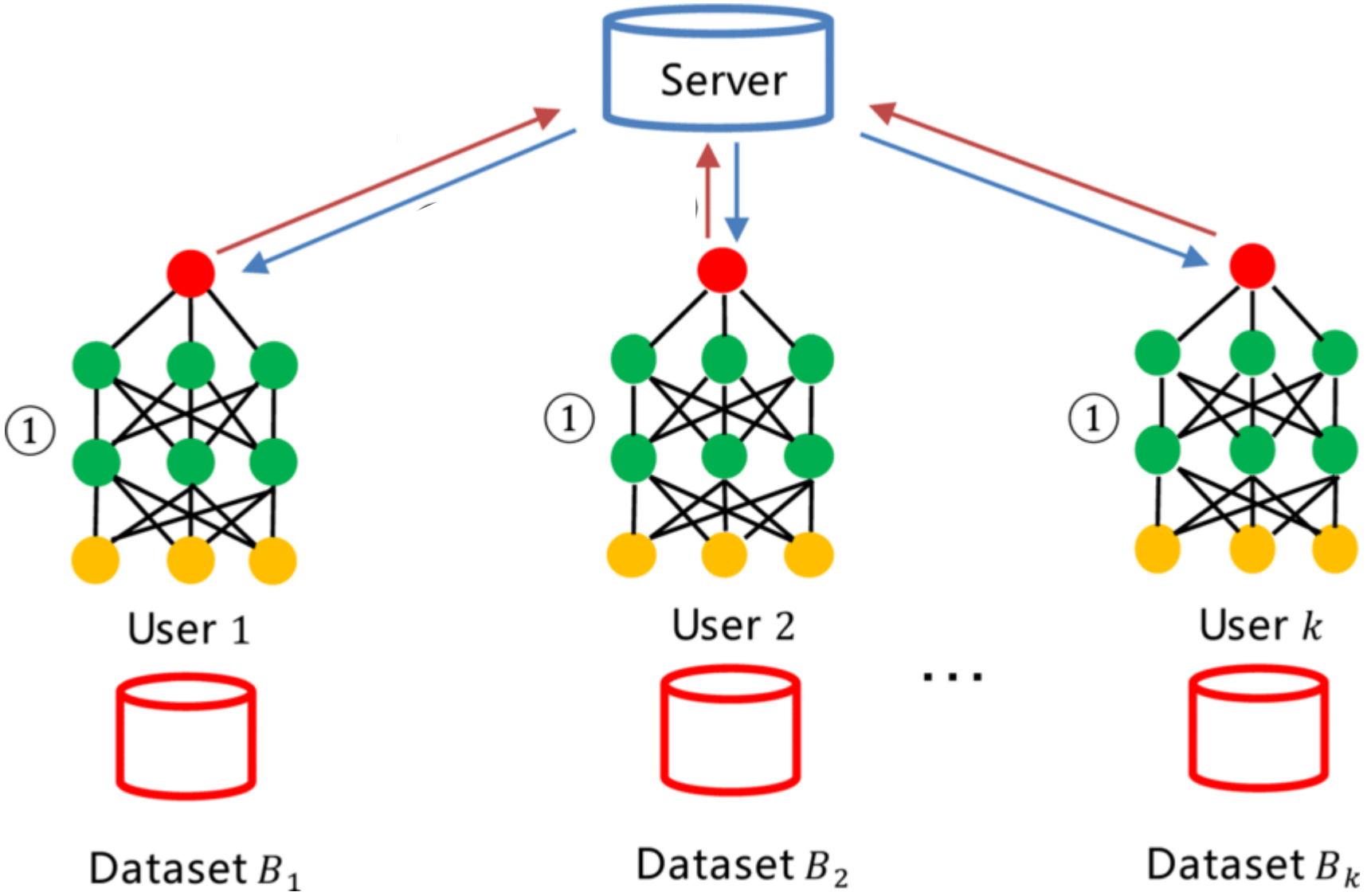
IV. SITUA-CQ algorithm

V. Result

What is Federated Learning?

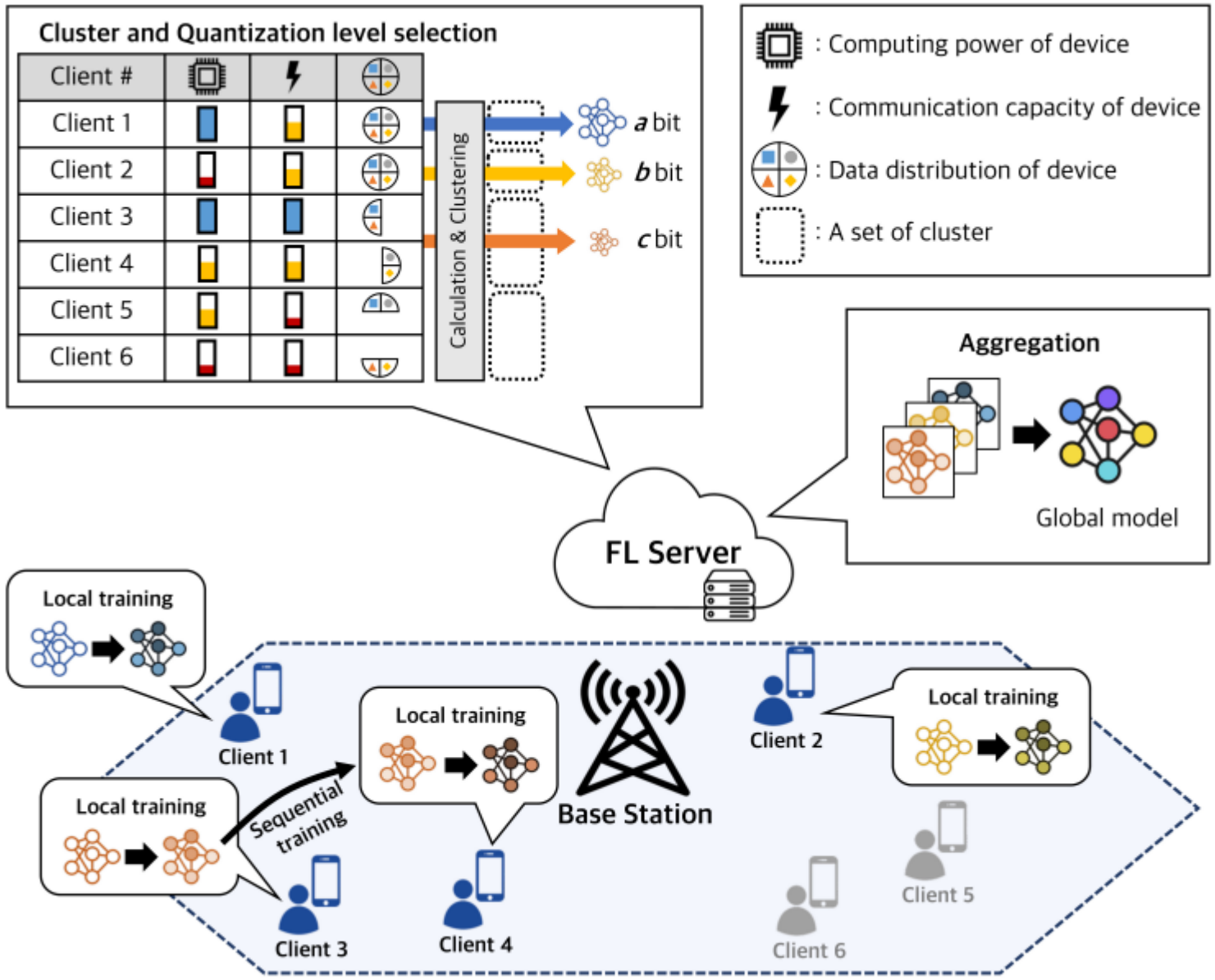


What is Federated Learning?



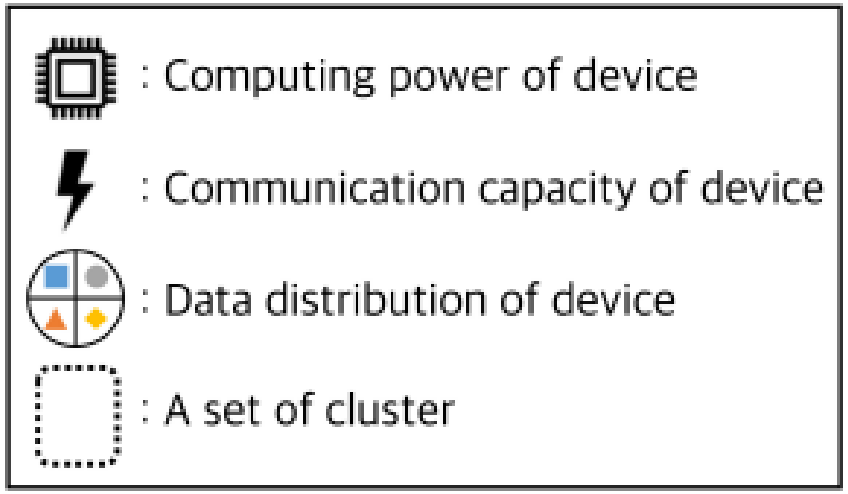
System model

System process

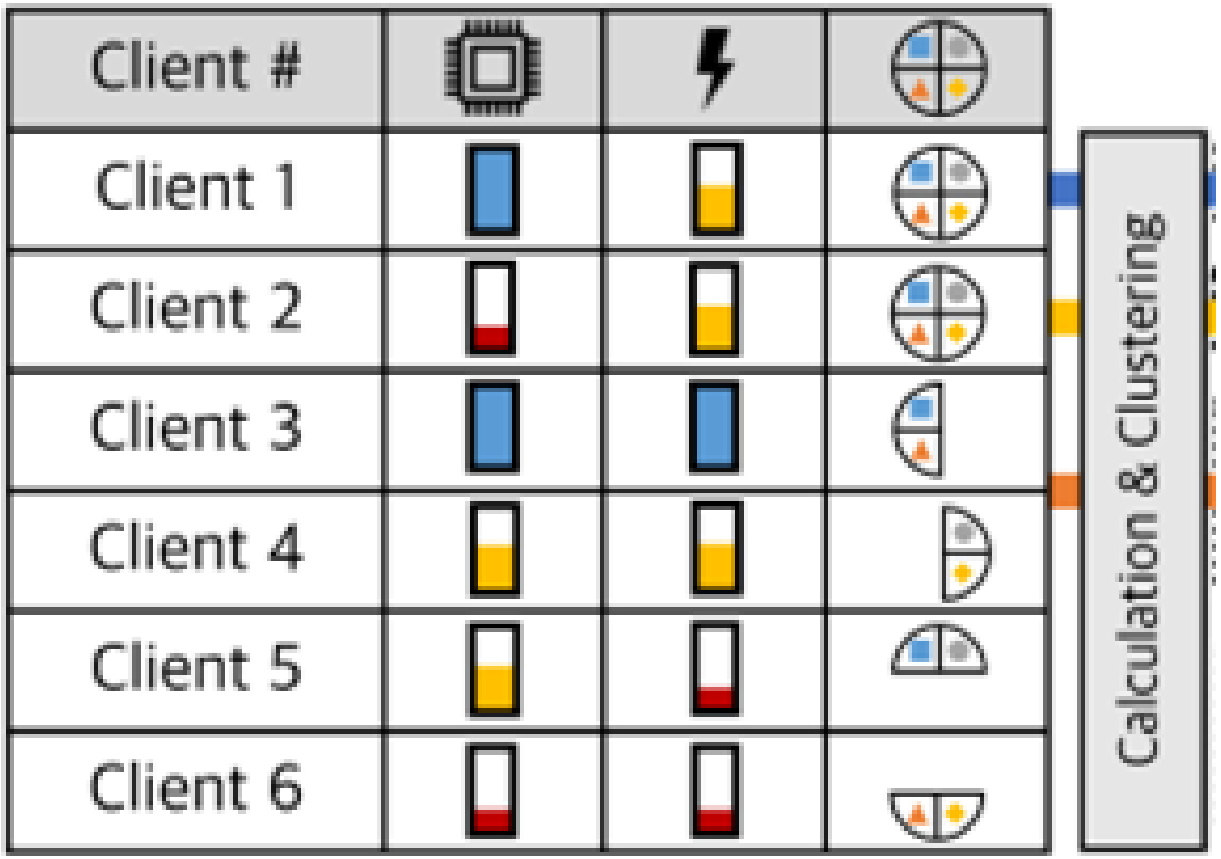


1. Performance reporting

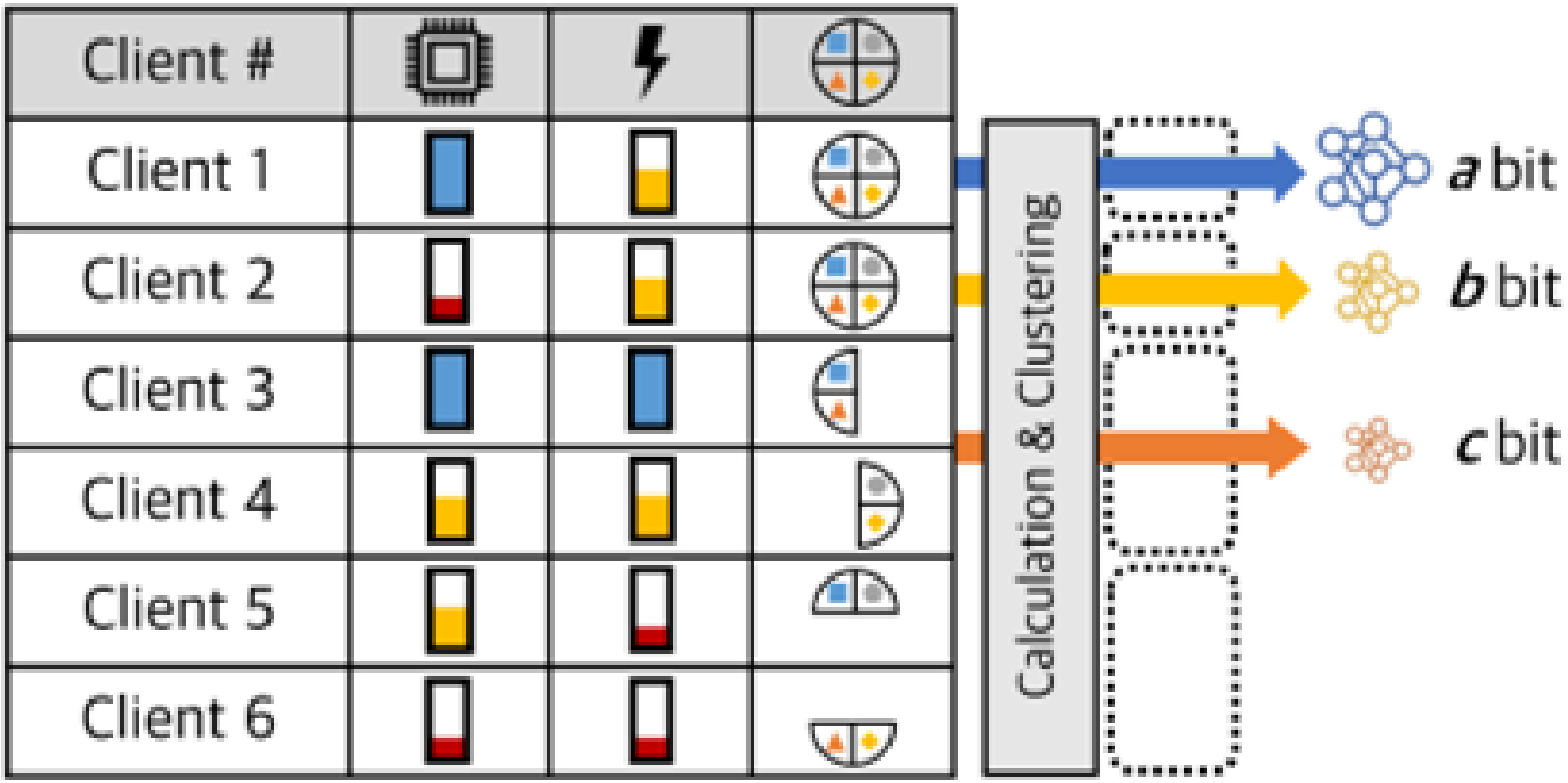
Client #			
Client 1			
Client 2			
Client 3			
Client 4			
Client 5			
Client 6			



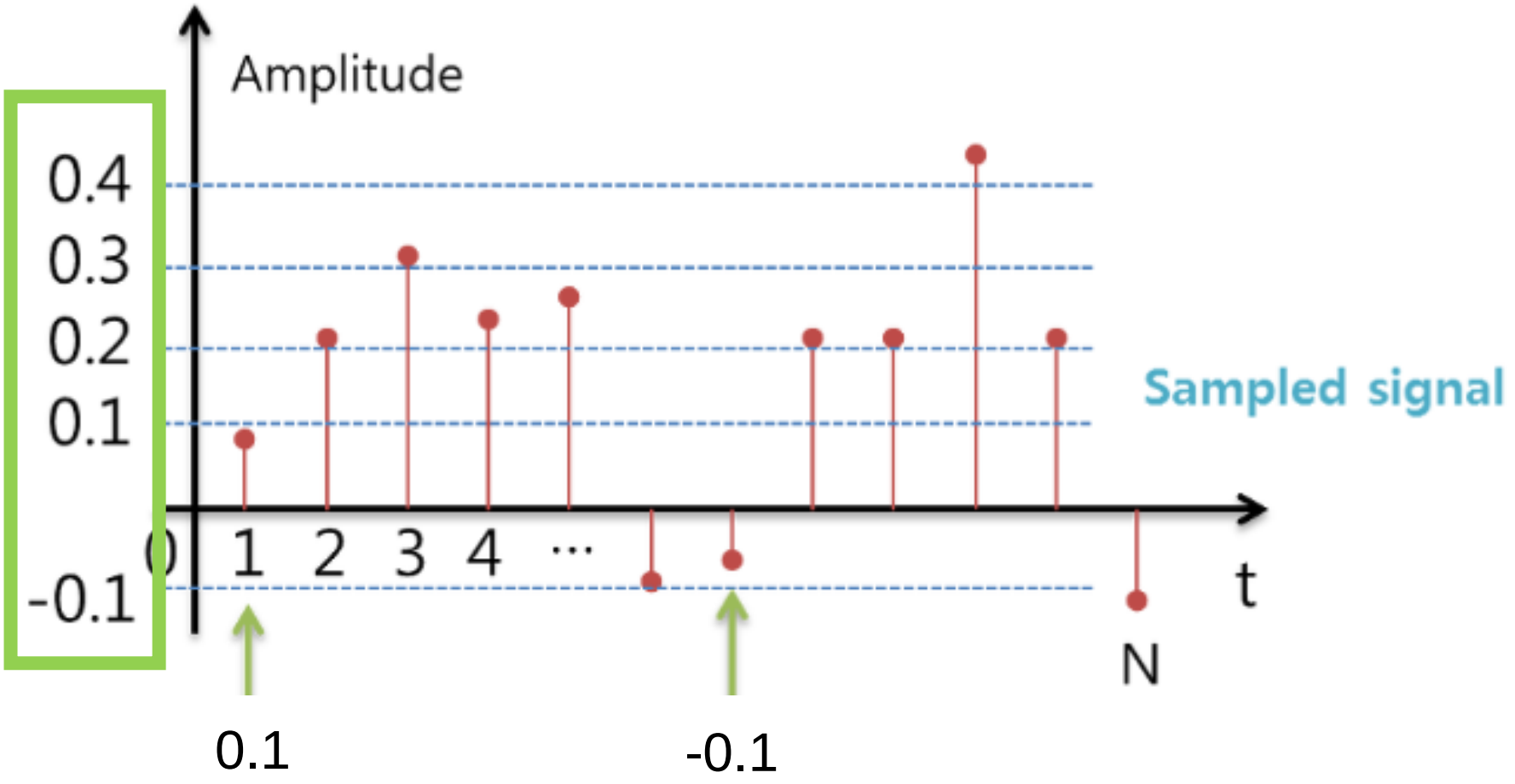
2. Cluster
3. Round time estimate



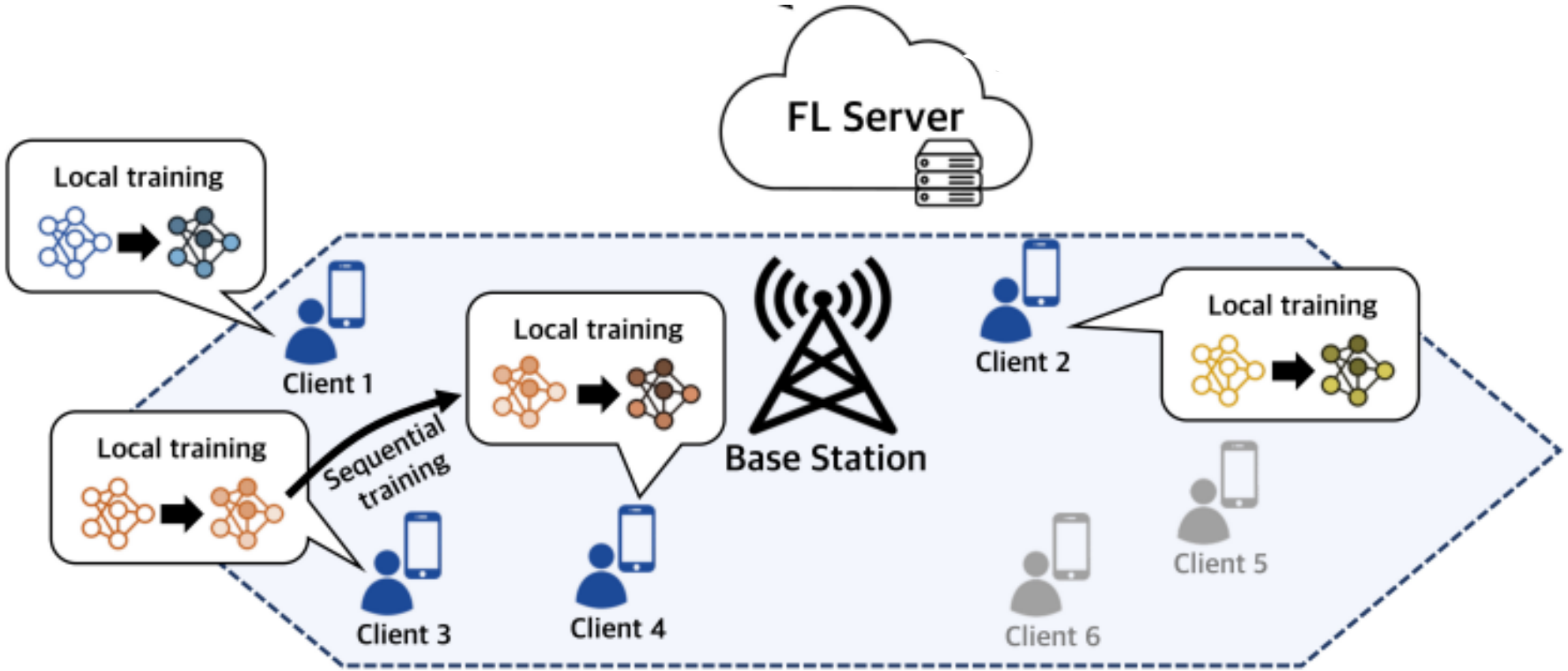
4. Quantization level section



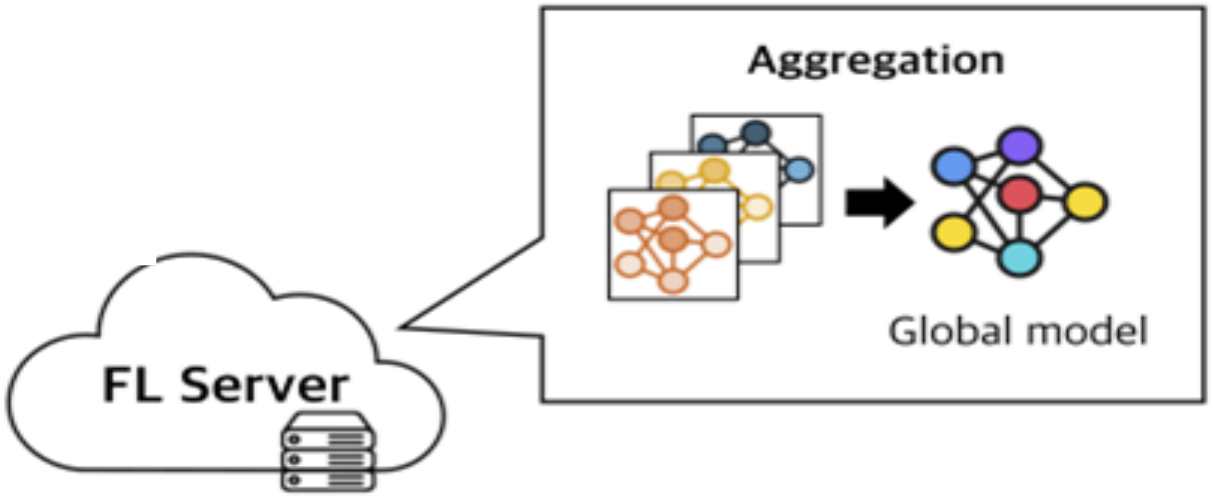
Quantization level?



5. Local training



5. Aggregation



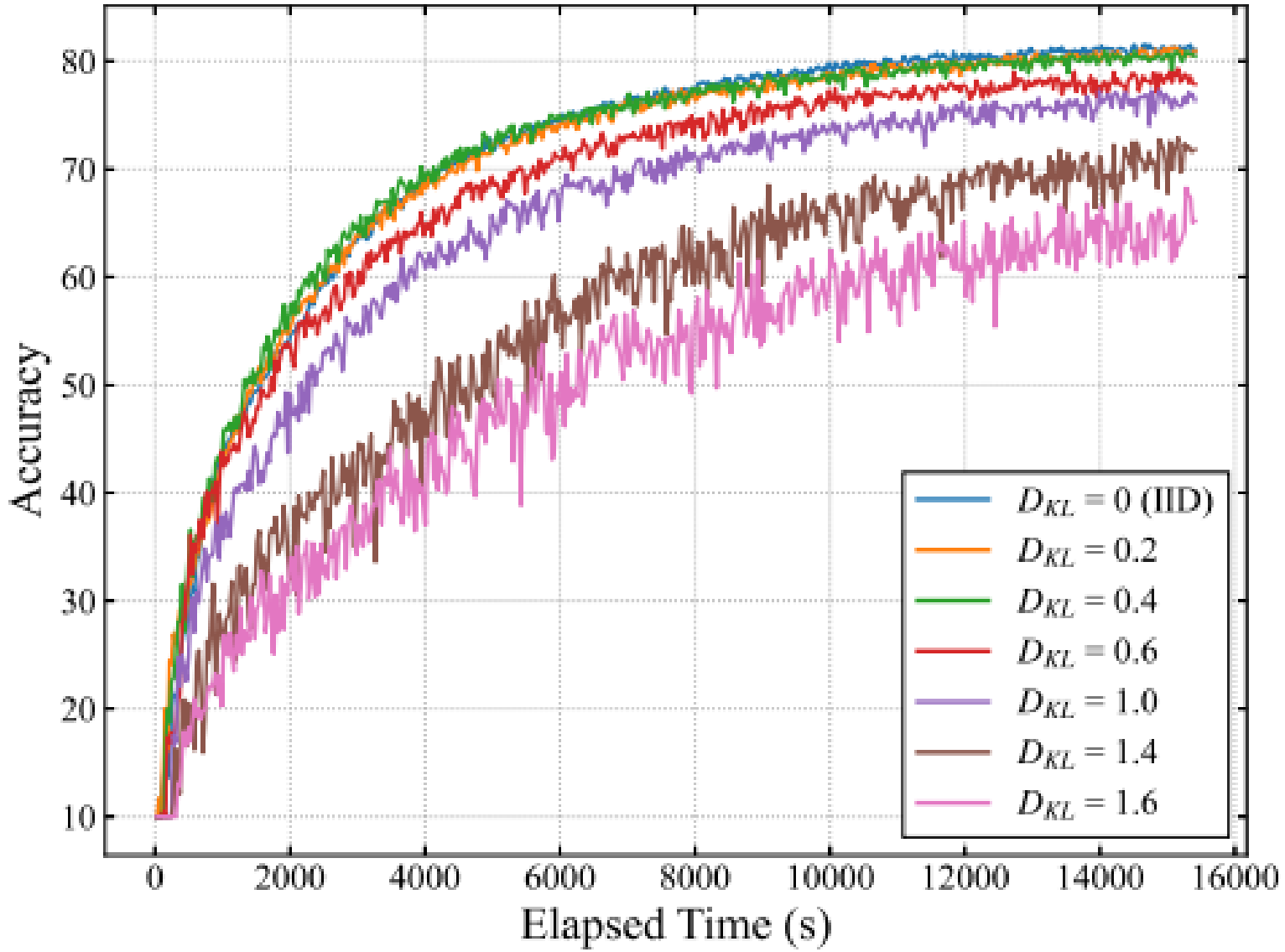
Deep model

```
class CNNQuant(nn.Module):
    def __init__(self,model_name,quant):
        super(CNNQuant,self).__init__()
        self.quant = quant()
        self.conv_1 = nn.Conv2d(3,32,kernel_size=3,padding=1)
        self.conv_2 = nn.Conv2d(32,32,kernel_size=3,padding=1)
        self.conv_3 = nn.Conv2d(32,64,kernel_size=3,padding=1)
        self.conv_4 = nn.Conv2d(64,64,kernel_size=3,padding=1)
        self.conv_5 = nn.Conv2d(64,128,kernel_size=3,padding=1)
        self.conv_6 = nn.Conv2d(128,128,kernel_size=3,padding=1)
        self.relu = nn.ReLU(inplace=True)
        self.softmax = nn.Softmax()
        self.dropout=nn.Dropout(p=0.5)
        self.maxpool=nn.MaxPool2d(kernel_size=2,stride=2)
        self.features = self._make_layers(cfg[model_name])
        self.flatten = nn.Linear(2048,512)
        self.flatten2 = nn.Linear(4096,512)
        self.classifier = nn.Linear(512, 10)
        self.classifier2 = nn.Linear(512, 10)
```

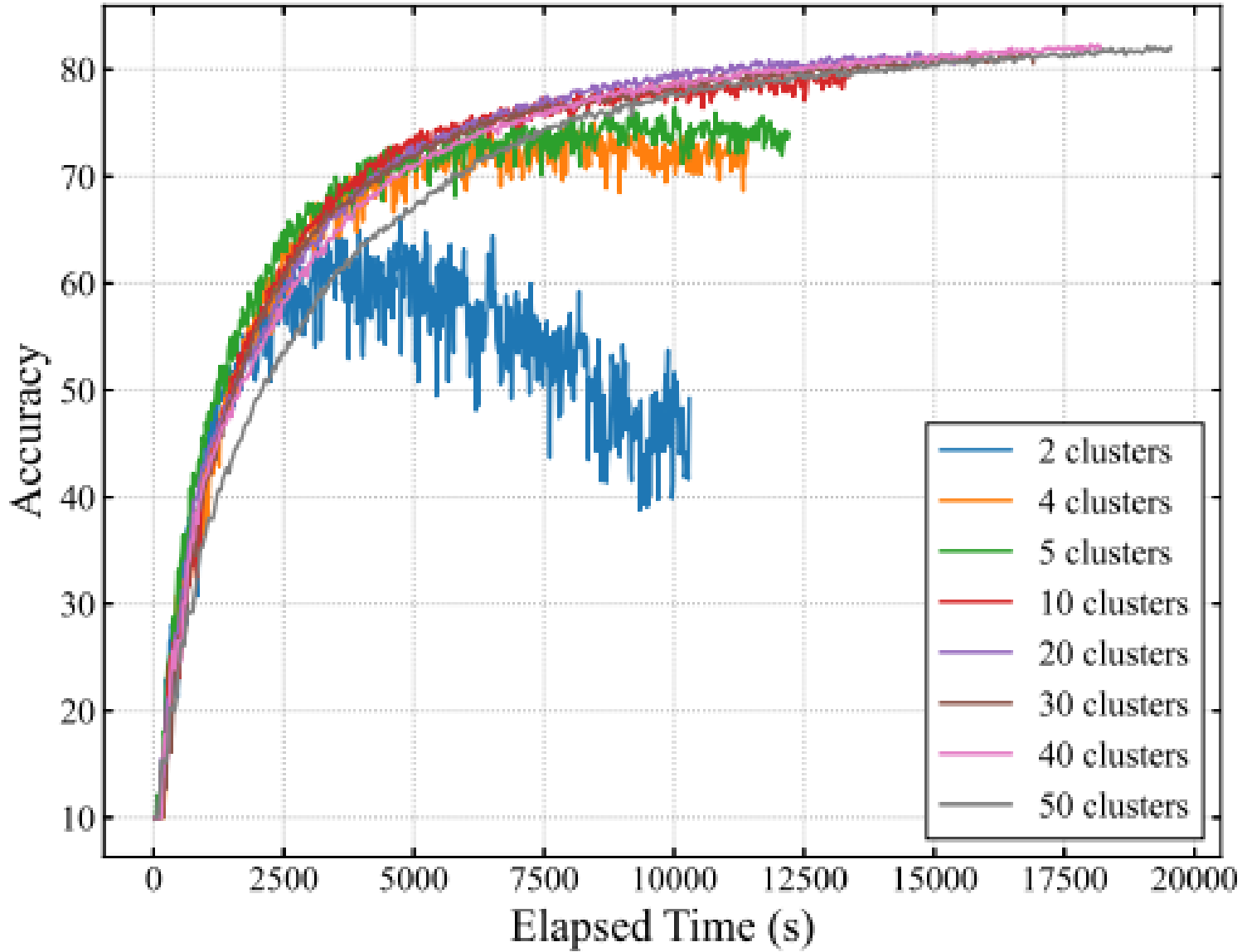
6 layers
32, 64, 126 filters

Relu, Max-pooling

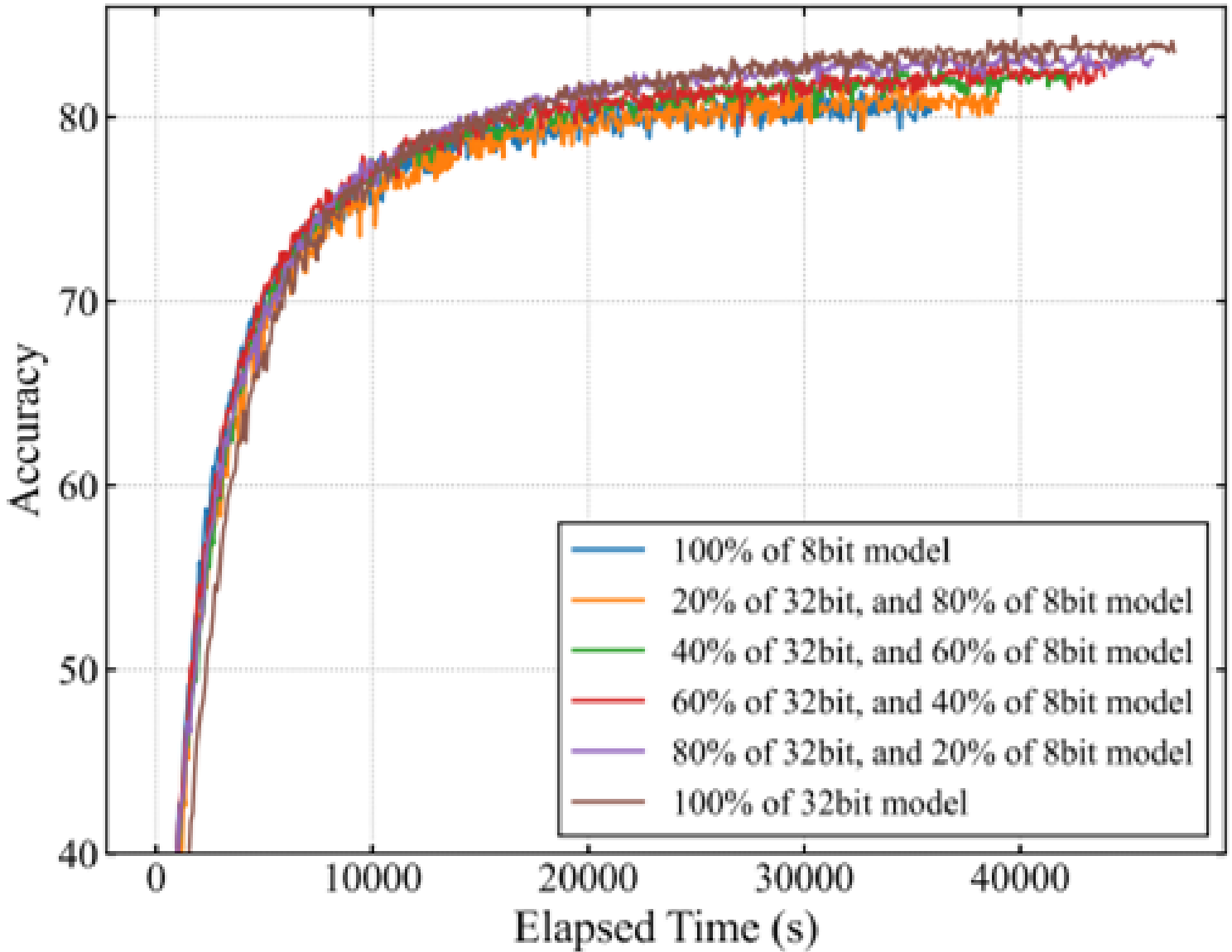
Effect data distribution of clusters on the accuracy



Effect of the number of selected clusters



Effect of the distribution of quantization levels



SITUA-CQ

Algorithm 1 SITUA-CQ

Initialize: the sets S_j, U_C, A_S, U_P, and U_B
Receive performance information from clients
$j \leftarrow 0$, $k \leftarrow 0$
$z_{j,b,t} \leftarrow 0, \forall j, b, t$
while $U_P \neq \emptyset$ **do**
  $k^* \leftarrow \arg\min_{k \in U_P} D_{KL}((P_{S_j} + P_k)||P_T)$
  Add k^* to S_j
  Delete k^* from U_P
  if $D_{KL}(P_{S_j}||P_T) \leq \theta_d$ **then**
    Add set S_j to U_C
    $j \leftarrow j + 1$
  end if
end while
$l \leftarrow 0$
while $l \leq \theta_k$ **do**
  $j^* \leftarrow \arg\min_{S_j \in U_C} R_{j,B,t}$
  $y_{j^*,k,t} = 1, \ \forall k \in S_{j^*}$
  Add S_{j^*} to A_S
  Remove S_{j^*} from U_C
  $l \leftarrow l + |S_{j^*}|$
end while
$m \leftarrow 0$, $b^* \leftarrow \arg\max_{b \in U_B} b$
Remove b^* from U_B
while $m \leq |A_S|$ **do**
  $j^* \leftarrow \arg\min_{S_j \in A_S} R_{j,b^*,t}$, **s.t.** $\sum_{b \in U_B} z_{j^*,b,t} = 0$
  $z_{m,b^*,t} = 1$
  if $\sum_{n=b^*}^{B} \psi_{A_S,n,t} \geq \sum_{n=b^*}^{B} \theta_n$ **then**
    $b^* \leftarrow \arg\max_{b \in U_B} b$
    Remove b^* from U_B
  end if
  $m \leftarrow m + 1$
end while

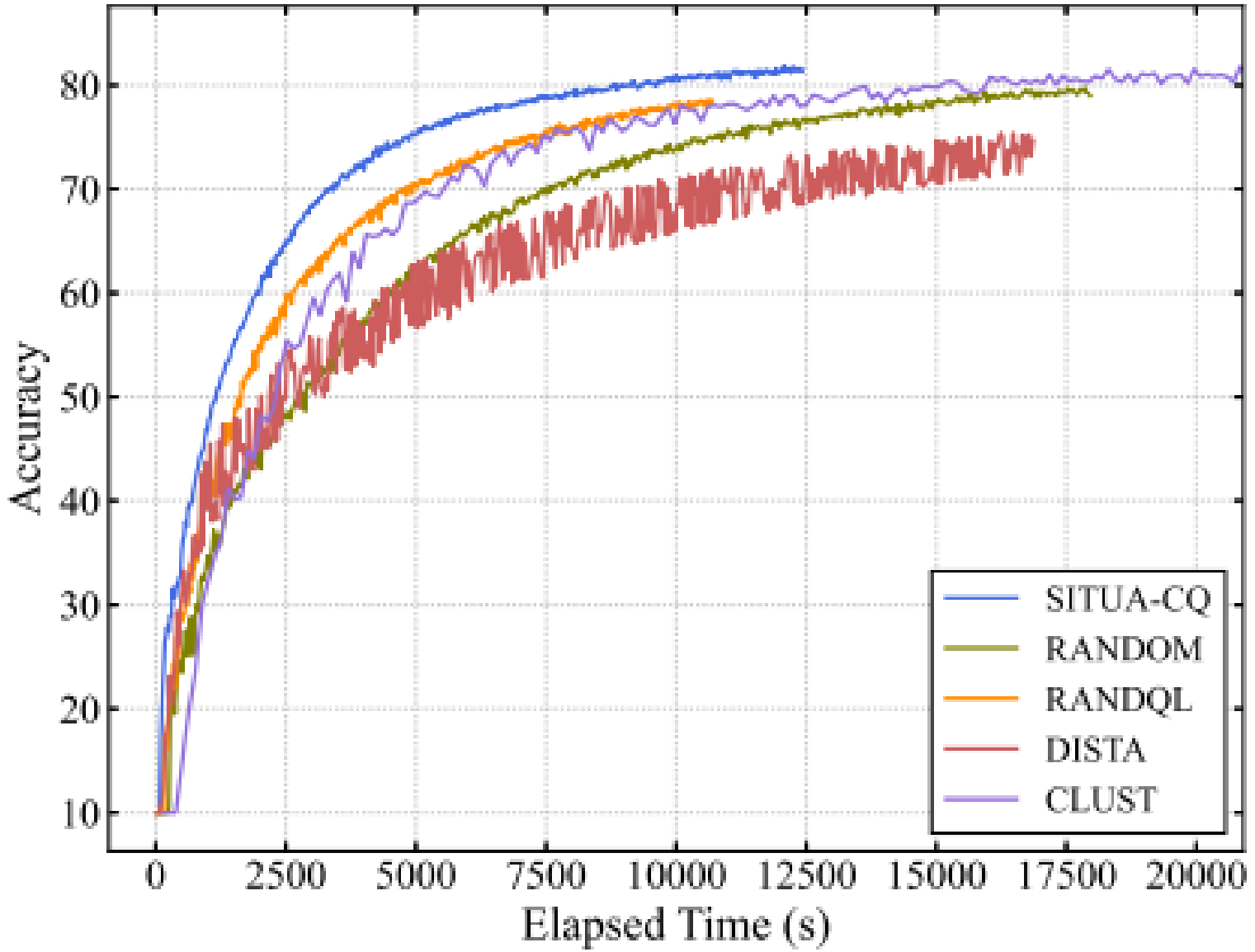
Cluster clients according data distribution of clients

Chose cluster with faster round time Until target number of clients

Allocate quantization level to selected cluster

Result

Accuracy curve of SITUA-CQ




Thank you