

An aerial view of a city skyline is shown in a dark, semi-transparent style. Overlaid on this background is a complex network of white lines connecting various nodes. Several circular icons are placed at key nodes: a Wi-Fi symbol in the top right, an envelope icon in the bottom left, a computer monitor icon in the middle right, and a shopping cart icon in the bottom right. Concentric circles emanate from some of these nodes, suggesting signal transmission or data flow.

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



**SITUA-CQ Algorithm**



**Federated Learning**



# **Abstract**

I. Federated Learning

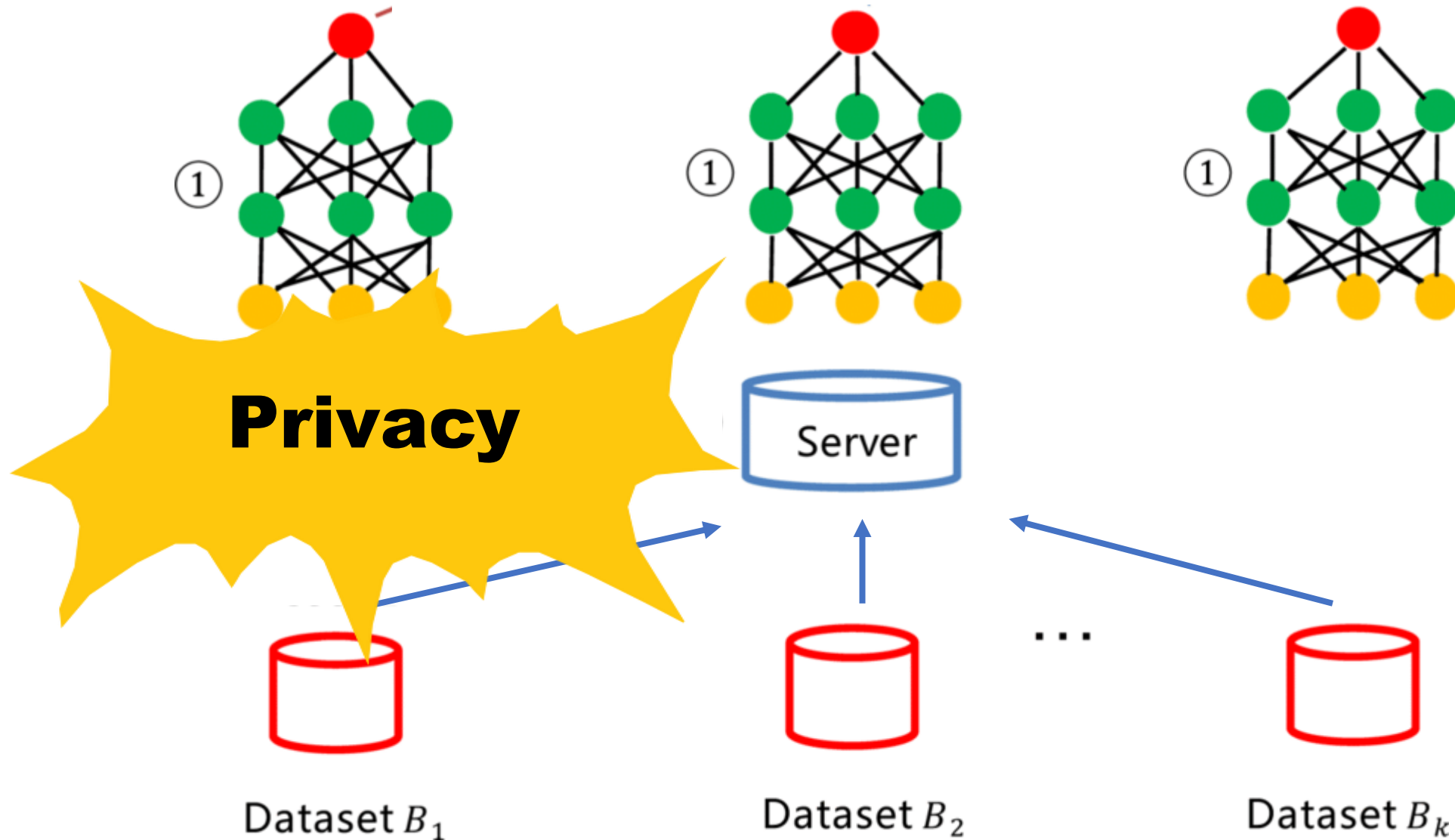
II. System model

III. Accuracy

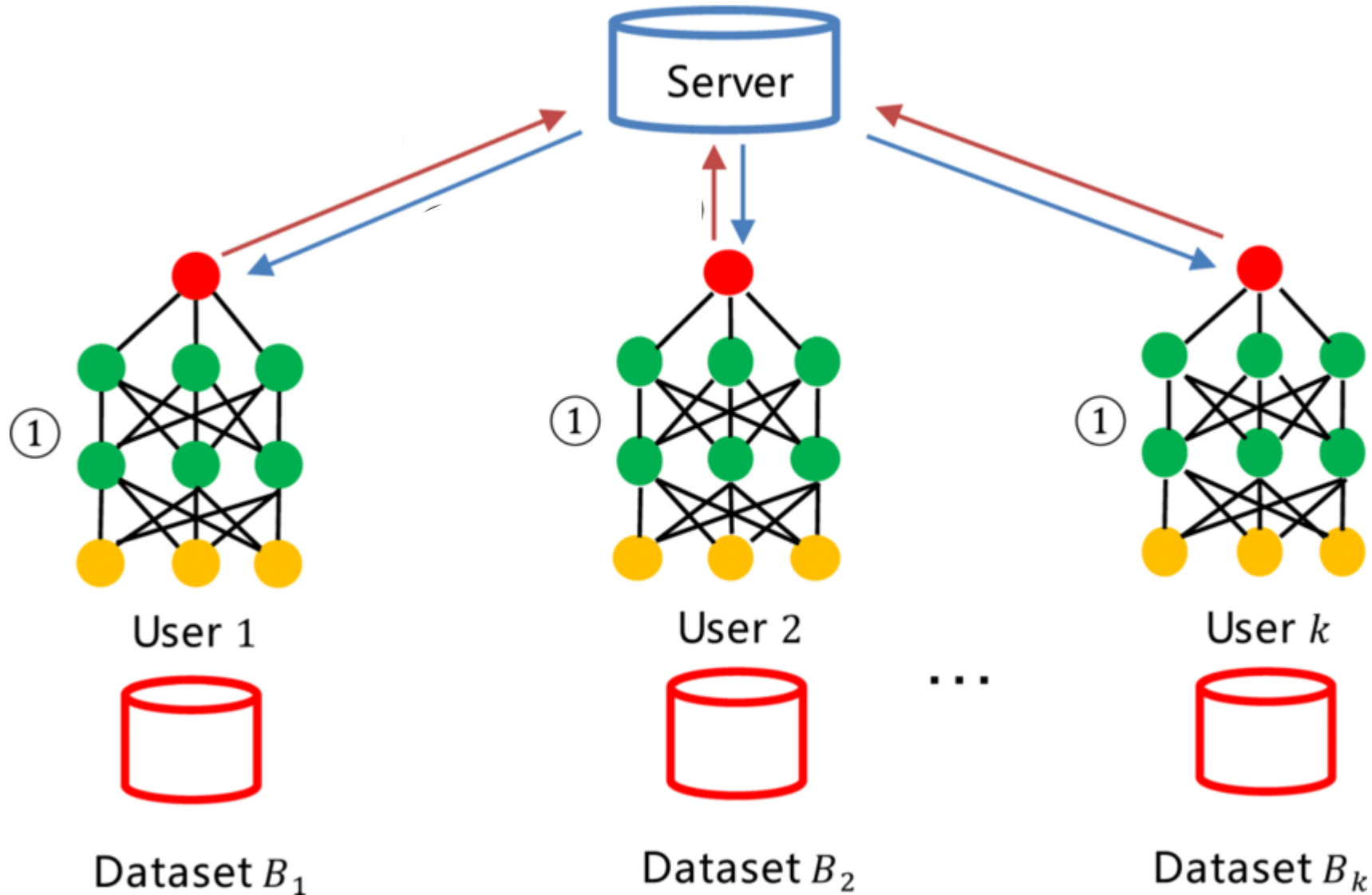
IV. SITUA-CQ algorithm

V. Result

# What is Federated Learning?



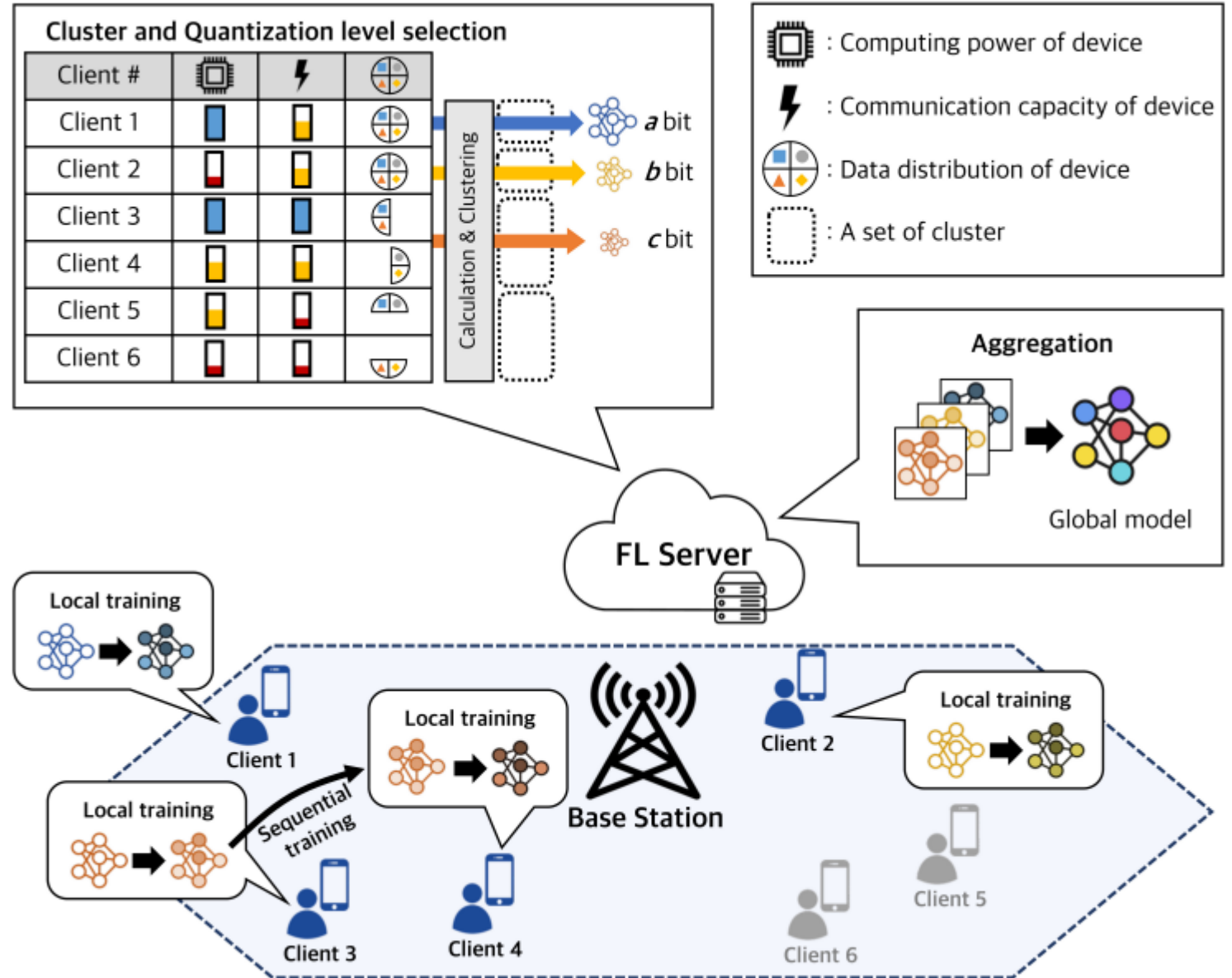
# What is Federated Learning?





# System model

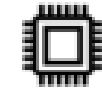
System process





# 1. Performance reporting

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



: Computing power of device



: Communication capacity of device



: Data distribution of device



: A set of cluster



## 2. Cluster

## 3. Round time estimate

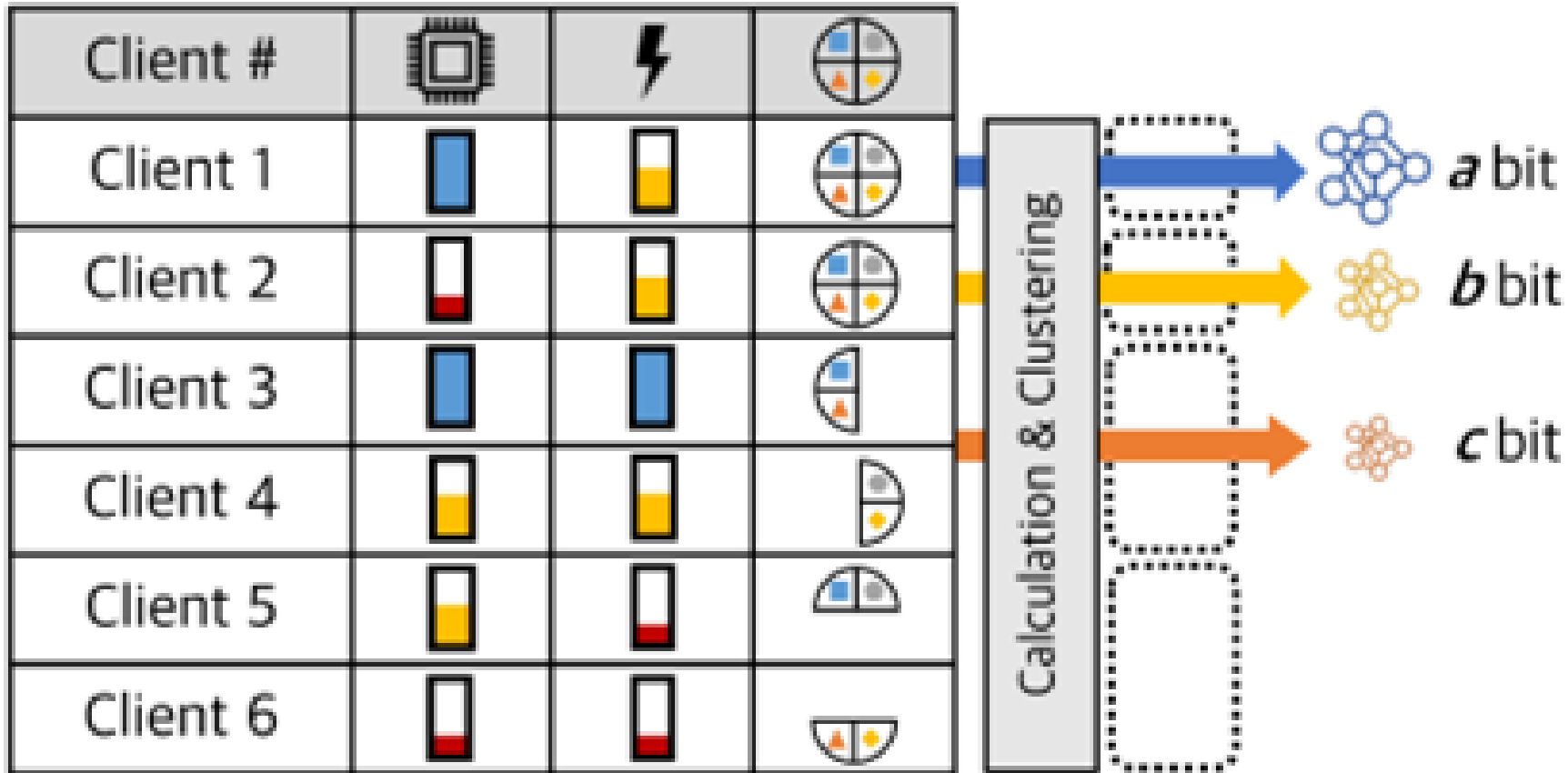
Client #			
Client 1			
Client 2			
Client 3			
Client 4			
Client 5			
Client 6			

Calculation & Clustering

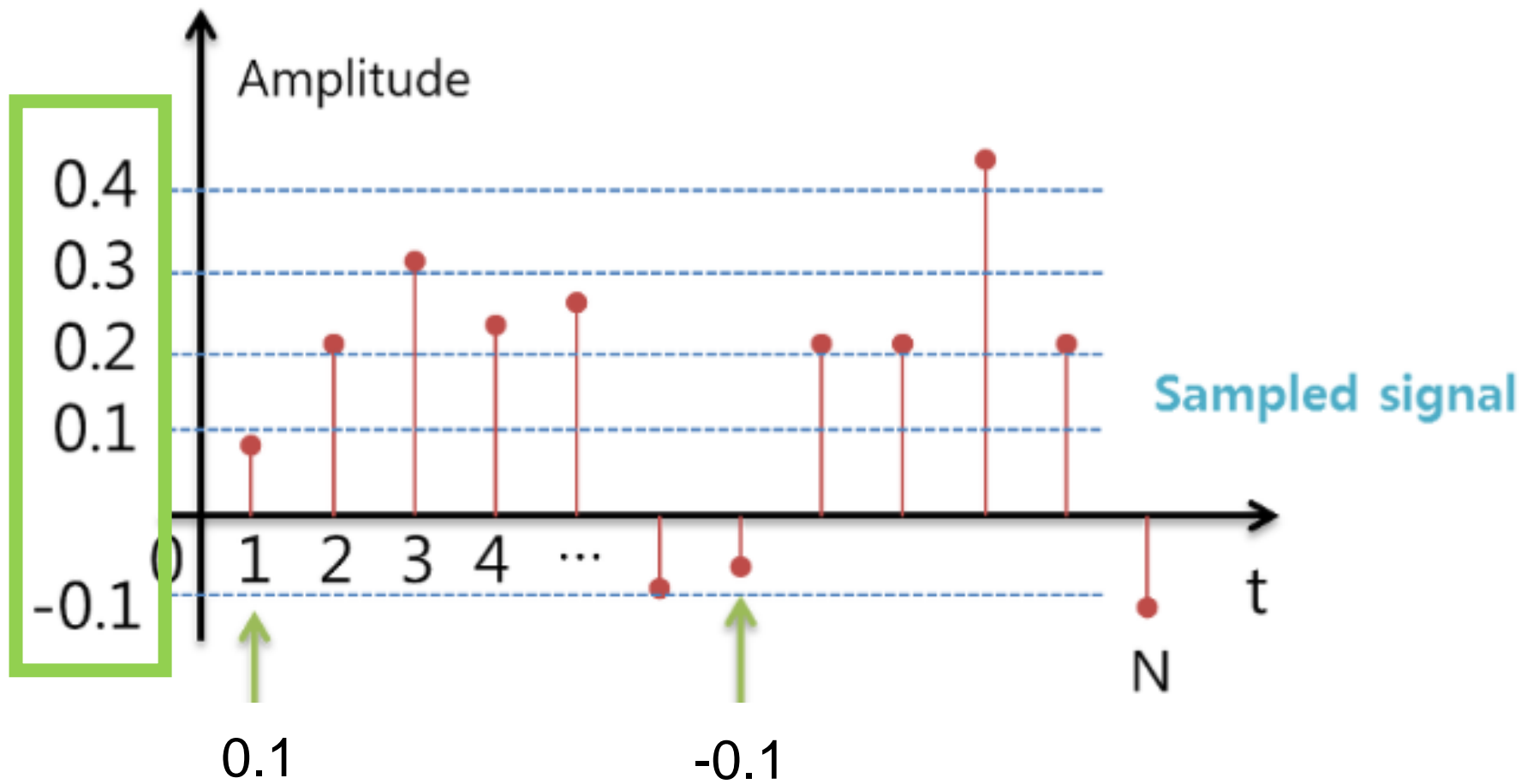




## 4. Quantization level section

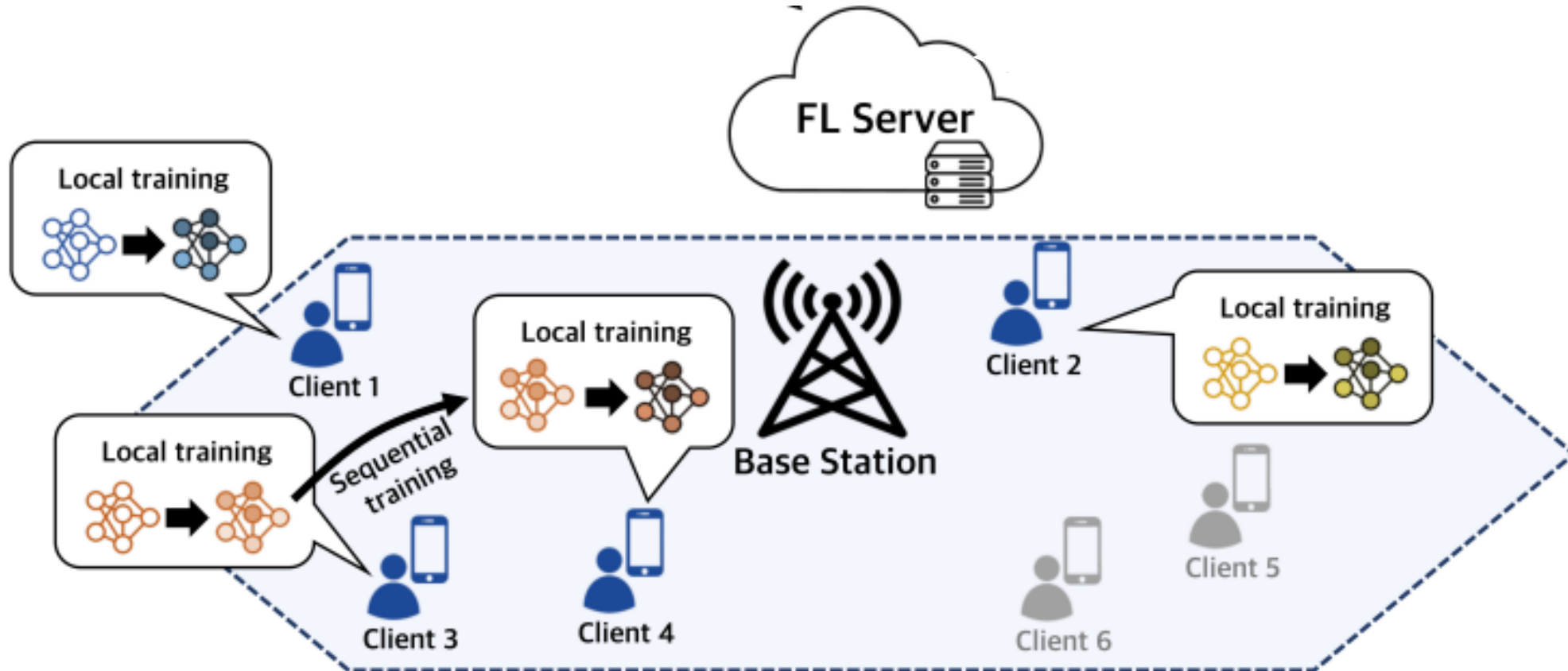


## Quantization level?



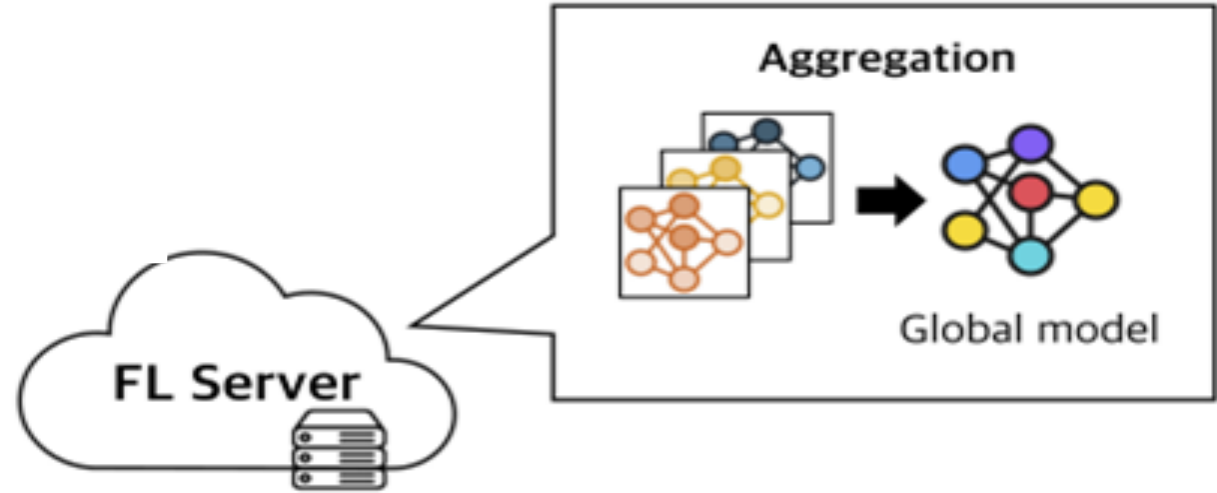


## 5. Local training





## 5. Aggregation





# Deep model

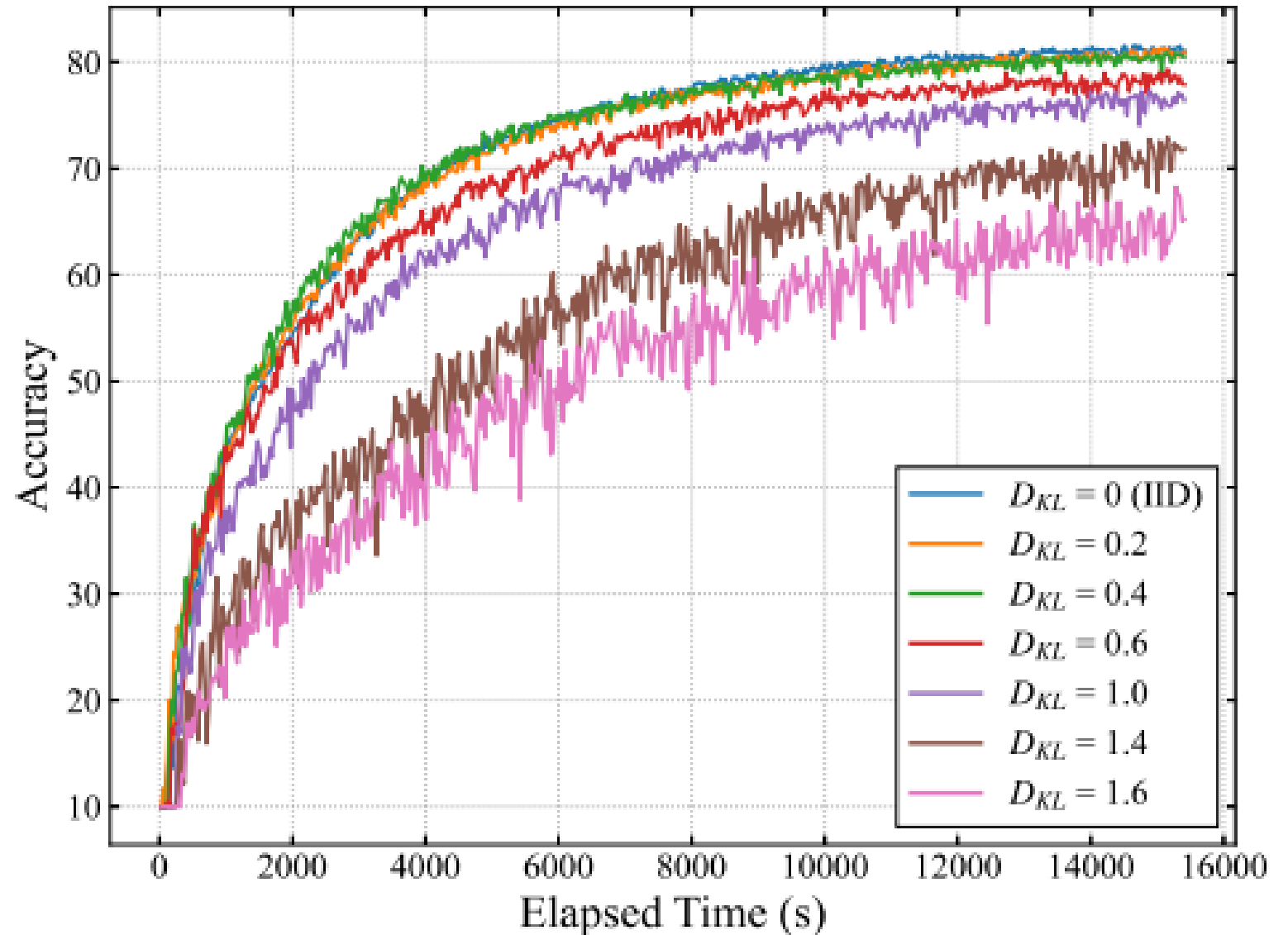
```
class CNNQuant(nn.Module):
    def __init__(self,model_name,quant):
        super(CNNQuant,self).__init__()
        self.quant = quant()
        self.conv_1 = nn.Conv2d(3,32,kernel_size=3,padding=1)
        self.conv_2 = nn.Conv2d(32,32,kernel_size=3,padding=1)
        self.conv_3 = nn.Conv2d(32,64,kernel_size=3,padding=1)
        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,stroke=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

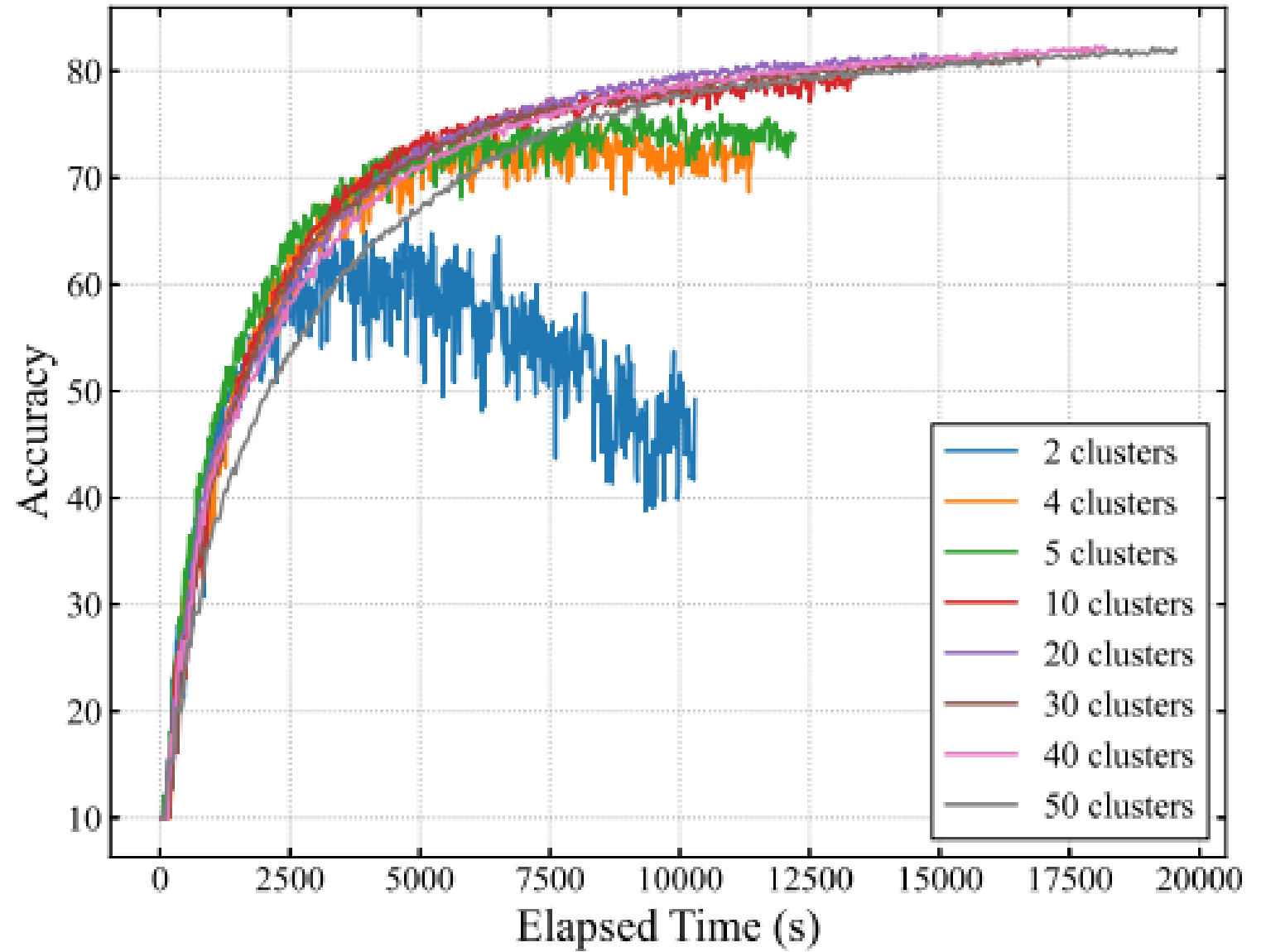
Relu, Max-  
pooling



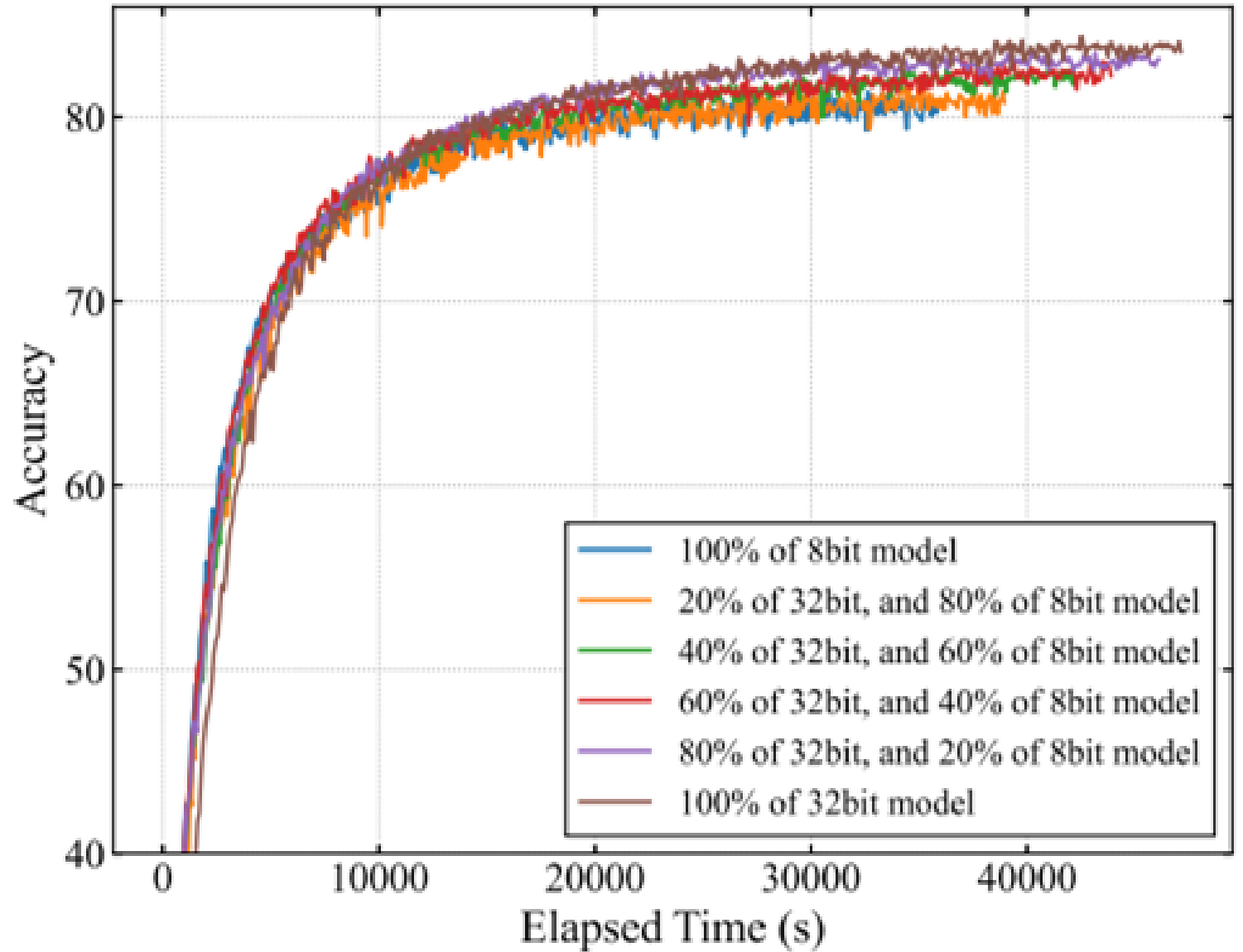
# Effect data distribution of clusters on the accuracy



# Effect of the number of selected clusters



# Effect of the distribution of quantization levels





# SITUA-CQ

## Algorithm 1 SITUA-CQ

```
1: Initialize: the sets  $S_j$ ,  $U_C$ ,  $A_S$ ,  $U_P$ , and  $U_B$ 
2: Receive performance information from clients
3:  $j \leftarrow 0$ ,  $k \leftarrow 0$ 
4:  $z_{j,b,t} \leftarrow 0, \forall j, b, t$ 
5: while  $U_P \neq \emptyset$  do
6:    $k^* \leftarrow \arg \min_{k \in U_P} D_{KL}((P_{S_j} + P_k) || P_T)$ 
7:   Add  $k^*$  to  $S_j$ 
8:   Delete  $k^*$  from  $U_P$ 
9:   if  $D_{KL}(P_{S_j} || P_T) \leq \theta_d$  then
10:     Add set  $S_j$  to  $U_C$ 
11:      $j \leftarrow j + 1$ 
12:   end if
13: end while
14:  $l \leftarrow 0$ 
15: while  $l \leq \theta_k$  do
16:    $j^* \leftarrow \arg \min_{S_j \in U_C} R_{j,B,t}$ 
17:    $y_{j^*,k,t} = 1, \forall k \in S_{j^*}$ 
18:   Add  $S_{j^*}$  to  $A_S$ 
19:   Remove  $S_{j^*}$  from  $U_C$ 
20:    $l \leftarrow l + |S_{j^*}|$ 
21: end while
22:  $m \leftarrow 0$ ,  $b^* \leftarrow \arg \max_{b \in U_B} \theta$ 
23: Remove  $b^*$  from  $U_B$ 
24: while  $m \leq |A_S|$  do
25:    $j^* \leftarrow \arg \min_{S_j \in A_S} R_{j,b^*,t}, \text{ s.t. } \sum_{b \in U_B} z_{j^*,b,t} = 0$ 
26:    $z_{m,b^*,t} = 1$ 
27:   if  $\sum_{n=b^*}^B \psi_{A_S,n,t} \geq \sum_{n=b^*}^B \theta_n$  then
28:      $b^* \leftarrow \arg \max_{b \in U_B} b$ 
29:     Remove  $b^*$  from  $U_B$ 
30:   end if
31:    $m \leftarrow m + 1$ 
32: end while
```

Cluster clients  
according data  
distribution of clients

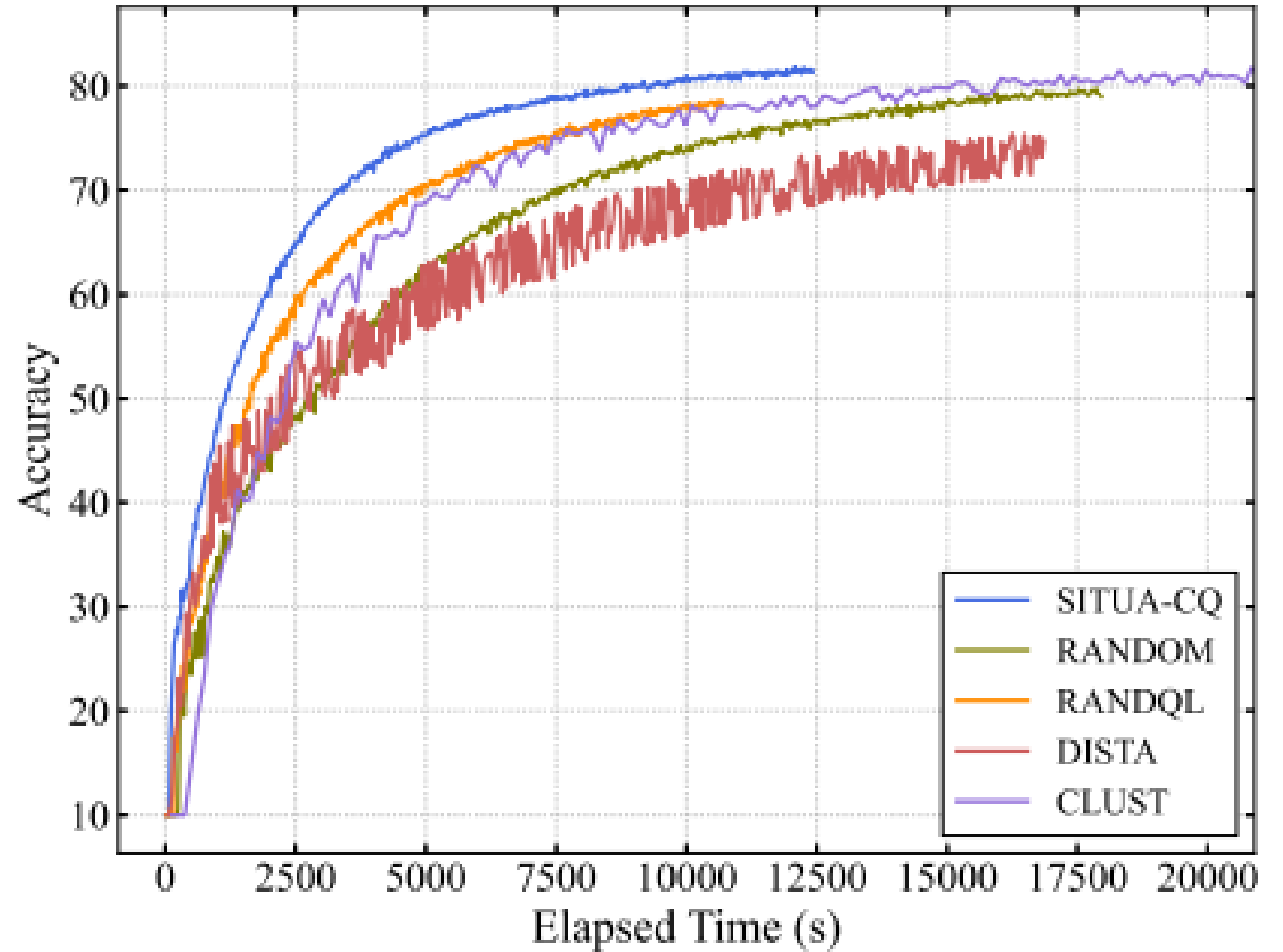
Chose cluster with  
faster round time  
Until target number of  
clients

Allocate quantization  
level to selected cluster



# Result

Accuracy curve of SITUA-CQ





# Thank you