Supplementary Information for "Introducing Edge Intelligence to Smart Meters via Federated Split Learning"

Yehui Li[1†], Dalin Qin[1†], H. Vincent Poor[2*], Yi Wang[1*]

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China.
[2]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, USA.

*Corresponding author(s). E-mail(s): poor@princeton.edu; yiwang@eee.hku.hk;
Contributing authors: yhli@eee.hku.hk; dlqin@eee.hku.hk;
†These authors contributed equally to this work.

Supplementary Notes

Supplementary Note 1: Experimental setting

In our experiments, L2 loss is adopted for both the label loss and knowledge distillation loss. The models are trained using the Adam optimizer with an initial learning rate of 5e-4. The mini-batch size is set to 32. The weights of label loss and knowledge distillation loss are both set to 0.5. To simulate the device heterogeneity in a real smart grid, 30 microcontrollers are randomly set to different operating frequencies between 21MHz and 84MHz (see Supplementary Fig. 7). In synchronous aggregation, the global training rounds for each experiment are fixed at 100 rounds. In asynchronous aggregation methods, the deviation of a single cluster gradient makes the training rounds longer for convergence, so we manually chose the number of training rounds at different numbers of clusters based on the loss of global training to ensure global model convergence (see Supplementary Fig. 8). Note that a local fine-tuning personalization strategy is incorporated into all federated-based methods for 30 rounds. Each experiment is repeated 5 times to eliminate the effect of randomness.

Supplementary Note 2: Evaluation metrics

Expressions of RMSE, MAPE, sMAPE, and MAE are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{2}$$

$$\text{sMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i + \hat{y}_i} \times 100\% \tag{3}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

where $\hat{y}_i$ and y_i are the i-th forecasted and real load values, respectively, and n is the number of samples.

Supplementary Note 3: Method description

The implementation details of the full method are as follows and also given in Algorithm 1 and Algorithm 2.

1. **Model splitting**: the cloud server first finds the optimal split ratio to split the initial model $\mathbf{w}^0$ into three parts $\mathbf{w}_e^0$, $\mathbf{w}_p^0$, and $\mathbf{w}_r^0$ and then distributes them to edge servers and smart meters, respectively. After that, each smart meter generates an auxiliary regressor $\mathbf{w}_a^0$.

2. **Model training**: In the round t, smart meters perform forward propagation for each sample on the feature extractor $\mathbf{w}_e^t$ and uploads the activation h_e to edge servers. Then h_e is sequentially forwarded as input through the feature processor $\mathbf{w}_p^t$ and the regressor $\mathbf{w}_r^t$, and the loss ℓ_s of the main model is calculated. Simultaneously, the auxiliary regressor $\mathbf{w}_a^t$ also propagates forward with h_e to obtain the prediction y_c, which is combined with the main model's prediction y_s to calculate the auxiliary model's distillation loss ℓ_c. Afterwards, smart meters calculate the gradient for $\mathbf{w}_r^t$ based on the error $\ell_{s,k}$. After receiving the gradient from smart meters, edge servers next calculate the gradient for $\mathbf{w}_p^t$. In parallel with the above process, smart meters calculate the gradients for $\mathbf{w}_a^t$ and $\mathbf{w}_e^t$ locally based on the error $\ell_{c,k}$. Eventually, the model parameters are updated based on the calculated gradients to complete a round of training.
3. **Model aggregation**: In the initialization phase, the cloud server designates smart meters to the edge servers to collaborate with by clustering their feature vectors V_k. After each smart meter finishes the model training in the t-th round, edge servers aggregate their parameters $\mathbf{w}_k^{t+1}$ to form the complete model $\mathbf{w}_{(i)}^{t+1}$. Finally, edge servers upload their aggregated model to update the global model $\mathbf{w}^{t+1}$. Cloud server distributes $\mathbf{w}^{t+1}$ to turn on the next round of model training.

Algorithm 1 Collaborative Split Learning

Function Model Splitting$(\mathbf{w}^0)$**:**

Choose optimal ratio α^* to split model: $\mathbf{w}^0 \rightarrow \mathbf{w}_e^0, \mathbf{w}_p^0, \mathbf{w}_r^0$

Cloud server allocates models to edge servers and smart meters

Smart meters generate auxiliary regressor $\mathbf{w}_a^0$

Function Model Training$(\mathbf{w}^t)$**:**

for *each* $x \in D$ **do**

`// Forward Propagation:`

$h_e \leftarrow f(\mathbf{w}_e^t, x)$

Smart meters send h_e to edge servers

$h_p \leftarrow f(\mathbf{w}_p^t, h_e)$

Edge servers return h_p to smart meters

$y_s \leftarrow f(\mathbf{w}_r^t, h_p)$ and $y_c \leftarrow f(\mathbf{w}_a^t, h_e)$

$\ell_s \leftarrow \ell(y, y_s)$ and $\ell_c \leftarrow \gamma\ell(y, y_c) + (1-\gamma)\ell(y_s, y_c)$

`// Backward Propagation:`

Smart meters calculate $\nabla_r \ell_s(\mathbf{w}_s^t, x)$ and send it to edge servers

do in parallel

Edge servers calculate $\nabla_p \ell_s(\mathbf{w}_s^t, x)$

do in parallel

Smart meters calculates $\nabla_a \ell_c(\mathbf{w}_c^t, x)$ and $\nabla_e \ell_c(\mathbf{w}_c^t, x)$

`// Parameter Update:`

$\mathbf{w}_r^{t+1} \leftarrow \mathbf{w}_r^t - \eta_t \nabla_r \mathcal{L}_s(\mathbf{w}_s^t)$

$\mathbf{w}_p^{t+1} \leftarrow \mathbf{w}_p^t - \eta_t \nabla_p \mathcal{L}_s(\mathbf{w}_s^t)$

$\mathbf{w}_a^{t+1} \leftarrow \mathbf{w}_a^t - \eta_t \nabla_a \mathcal{L}_c(\mathbf{w}_c^t)$

$\mathbf{w}_e^{t+1} \leftarrow \mathbf{w}_e^t - \eta_t \nabla_e \mathcal{L}_c(\mathbf{w}_c^t)$

$\mathbf{w}^{t+1} \leftarrow [\mathbf{w}_e^{t+1}, \mathbf{w}_p^{t+1}, \mathbf{w}_r^{t+1}, \mathbf{w}_a^{t+1}]$

Return $\mathbf{w}^{t+1}$

Algorithm 2 Semi-Asynchronous Federated Learning

`// Initialization:`

for *each* $k \in A$ **do in parallel**

Smart meters upload feature vector $V_k \leftarrow [\frac{1}{P_{ED}}, \frac{1}{R}]$ to cloud server

Cloud server designates smart meters to the i-th edge server: $A_i \leftarrow Cluster(V_k)$

$\mathbf{w}_k^0 \leftarrow$ Model Splitting $(\mathbf{w}^0)$

Function Model aggregation`():`

for *each round* $t = 1, 2, ..., T$ **do**

for *each* $i = 1, 2, ..., M$ **do in parallel**

for *each* $k = 1, 2, ..., K_i$ **do in parallel**

Downloads global model $\mathbf{w}^t$ from cloud server

$\mathbf{w}_k^t \leftarrow \mathbf{w}^t$

$\mathbf{w}_k^{t+1} \leftarrow$ Model Training $(\mathbf{w}_k^t)$

`// Synchronous aggregation at edge servers:`

Smart meters upload model $\mathbf{w}_k^{t+1}$ to edge servers

$\mathbf{w}_{(i)}^{t+1} \leftarrow \frac{1}{K_i} \sum_{k=1}^{K_i} \mathbf{w}_k^{t+1}$

`// Asynchronous aggregation at cloud server:`

Edge servers upload model $\mathbf{w}_{(i)}^{t+1}$ to cloud server

$\mathbf{w}^{t+1} = (1 - \tau_i)\mathbf{w}^t + \tau_i \mathbf{w}_{(i)}^{t+1}$

Supplementary Note 4: Short-term scheduling formulation

The objective of the short-term scheduling is to schedule the energy consumption of home appliances to help consumers reduce electricity costs based on forecasted load, which can be expressed as:

$$\min C = \sum_{t=1}^{\mathrm{T}} \lambda_{\mathrm{t}} \left(P_{\mathrm{t}}^{\mathrm{fcst}} + P_{\mathrm{t}}^{\mathrm{AC}} + P_{\mathrm{t}}^{\mathrm{EV}} + P_{\mathrm{t}}^{\mathrm{ESS}} - P_{\mathrm{t}}^{\mathrm{solar}} \right) \tag{5}$$

where T denotes the scheduling time scale; λ_t denotes the time-of-use electricity price; P_t^{fcst} denotes the forecasted load consumption; $P_{\mathrm{t}}^{\mathrm{AC}}$, $P_{\mathrm{t}}^{\mathrm{EV}}$, $P_{\mathrm{t}}^{\mathrm{ESS}}$, and $P_{\mathrm{t}}^{\mathrm{solar}}$ denote the power of air conditioner (AC), electric vehicle (EV), energy storage system (ESS), and solar panel, respectively. The positive and negative signs of $P_{\mathrm{t}}^{\mathrm{ESS}}$ correspond to the discharge and charging states of the ESS.

The feasibility constraints limit the operating power of appliances within a feasible range, which can be formulated as follows:

$$\begin{aligned} &P_{\min}^{\mathrm{AC}} \le P_{\mathrm{t}}^{\mathrm{AC}} \le P_{\max}^{\mathrm{AC}} \\ &P_{\min}^{\mathrm{EV}} \le P_{\mathrm{t}}^{\mathrm{EV}} \le P_{\max}^{\mathrm{EV}} \\ &-P_{\max}^{\mathrm{ESS}} \le P_{\mathrm{t}}^{\mathrm{ESS}} \le P_{\max}^{\mathrm{ESS}} \end{aligned} \tag{6}$$

where $P_{\min}^{\mathrm{AC}}$ and $P_{\max}^{\mathrm{AC}}$ denote the minimum and maximum operating power of the AC, respectively; $P_{\min}^{\mathrm{EV}}$ and $P_{\max}^{\mathrm{EV}}$ denote the minimum and maximum charging power of the EV, respectively; $P_{\max}^{\mathrm{ESS}}$ denote the maximum charging and discharging power of the ESS.

The thermal dynamics constraints restrict the indoor temperature within a comfortable range, which can be formulated as follows:

$$\begin{aligned} &T_{t+1}^{\mathrm{in}} = \varepsilon T_t^{\mathrm{in}} + (1-\varepsilon)\left(T_t^{\mathrm{out}} + \eta^{\mathrm{AC}} \cdot \lambda \cdot P_t^{\mathrm{AC}} \cdot \Delta T\right) \\ &T_{\min}^{\mathrm{in}} \le T_t^{\mathrm{in}} \le T_{\max}^{\mathrm{in}} \end{aligned} \tag{7}$$

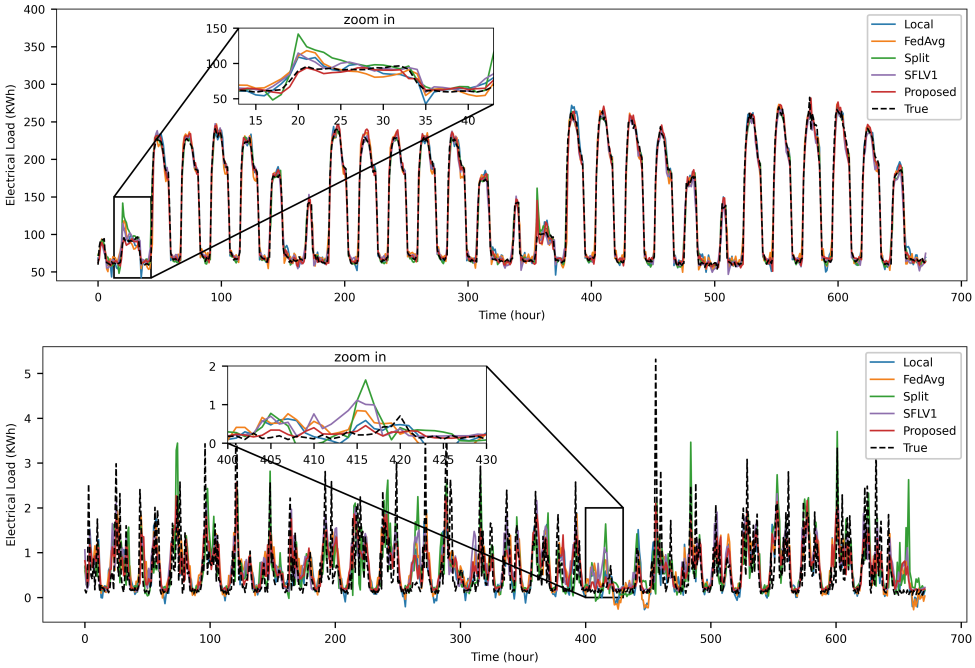
where T_t^{in} denotes the indoor temperature at time t; ε denotes the inertia factor; η^{AC} the thermal conversion efficiency; λ denotes the reciprocal of the thermal conductivity; ΔT denotes the scheduling resolution; $T_{\min}^{\mathrm{in}}$ and $T_{\max}^{\mathrm{in}}$ denote the minimum and maximum values of household preferred indoor temperatures, respectively.

The battery constraint restricts the temporal coupling of EV and ESS, which can be expressed as:

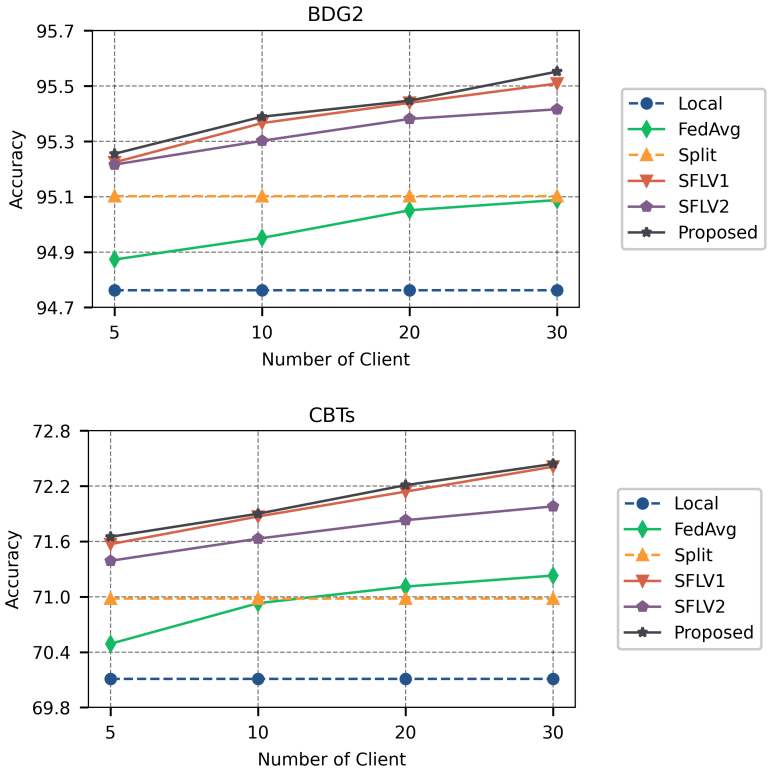
$$\begin{aligned} &\mathrm{SoC}_{t+1}^{\mathrm{EV}} = \mathrm{SoC}_t^{\mathrm{EV}} + \eta_{i,\mathrm{cha}}^{\mathrm{EV}} \cdot P_t^{\mathrm{EV}} \cdot \Delta T / C_{\max}^{\mathrm{EV}} \\ &\mathrm{SoC}_{t+1}^{\mathrm{ESS}} = \begin{cases} \mathrm{SoC}_t^{\mathrm{ESS}} + \eta_{i,\mathrm{cha}}^{\mathrm{ESS}} \cdot P_t^{\mathrm{ESS}} \cdot \Delta T / C_{\max}^{\mathrm{ESS}}, & P_t^{\mathrm{ESS}} \ge 0 \\ \mathrm{SoC}_{i,t}^{\mathrm{ESS}} + \eta_{i,\mathrm{dis}}^{\mathrm{ESS}} \cdot P_t^{\mathrm{ESS}} \cdot \Delta T / C_{\max}^{\mathrm{ESS}}, & P_t^{\mathrm{ESS}} < 0 \end{cases} \\ &\mathrm{SoC}_{\min}^{\mathrm{EV}} \le \mathrm{SoC}_t^{\mathrm{EV}} \le \mathrm{SoC}_{\max}^{\mathrm{EV}} \\ &\mathrm{SoC}_{\min}^{\mathrm{ESS}} \le \mathrm{SoC}_t^{\mathrm{ESS}} \le \mathrm{SoC}_{\max}^{\mathrm{ESS}} \end{aligned} \tag{8}$$

where $\mathrm{SoC}_{t+1}^{\mathrm{EV}}$ and $\mathrm{SoC}_{t+1}^{\mathrm{EV}}$ denote the state of charge (SoC) of the EV and ESS, respectively; $\eta_{i,\mathrm{cha}}$ denotes the charging efficiencies of the EV; $\eta_{i,\mathrm{cha}}$ and $\eta_{i,\mathrm{dis}}$ denote the charging and discharging efficiencies of the ESS, respectively; $C_{\max}^{\mathrm{EV}}$ and $C_{\max}^{\mathrm{ESS}}$ denotes the battery capacity of the EV and ESS, respectively; $\mathrm{SoC}_{\min}^{\mathrm{EV}}$ and $\mathrm{SoC}_{\max}^{\mathrm{EV}}$ denote the minimum and maximum SOC values of the EV, respectively; $\mathrm{SoC}_{\min}^{\mathrm{ESS}}$ and $\mathrm{SoC}_{\max}^{\mathrm{ESS}}$ denote the minimum and maximum SOC values of the ESS, respectively.

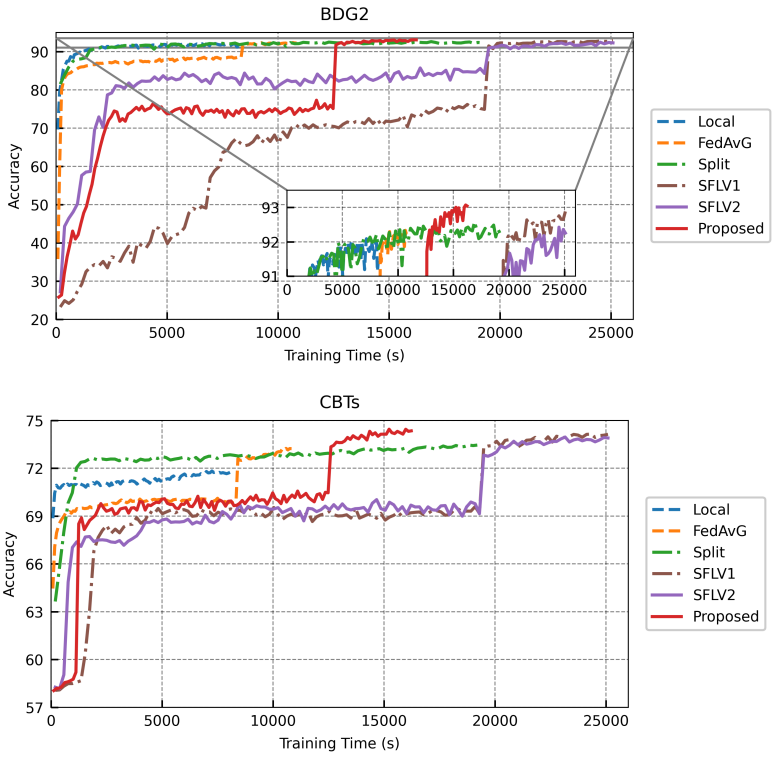
Supplementary Figures



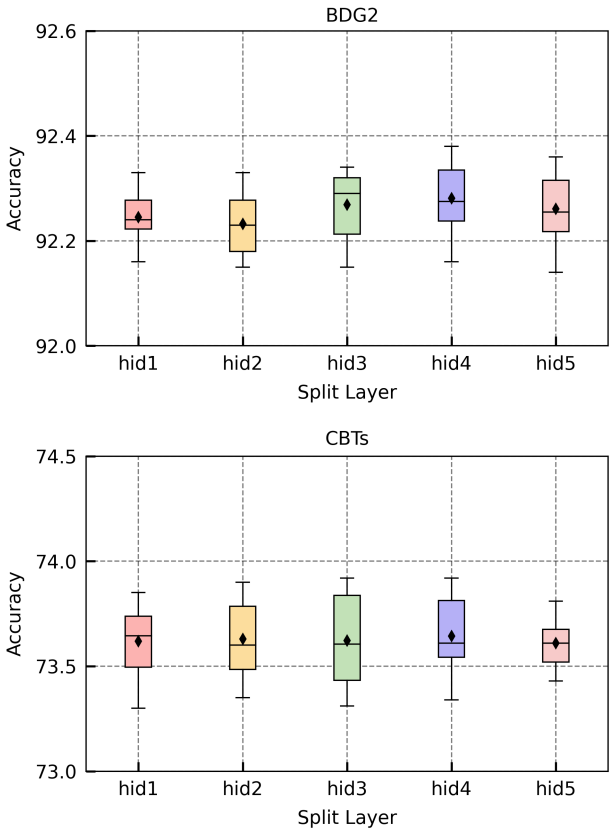
Supplementary Fig. 1: Comparison of forecasting results for six on-device feasible methods. We report the time series values of real and forecasting electrical load on datasets BDG2 and CBTs. The forecasting values of the proposed method are closest to the true values, especially in the case of high load volatility. Source data are provided as a Source Data file.



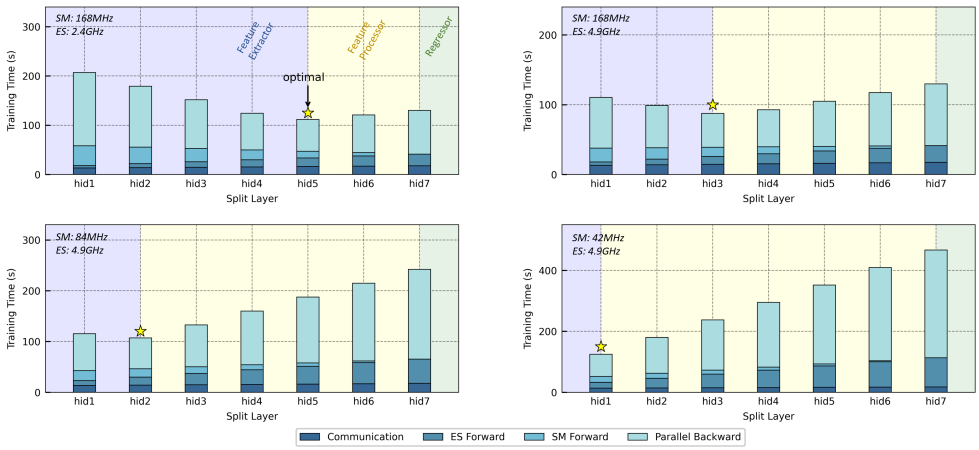
Supplementary Fig. 2: Performance evaluation of six on-device feasible methods on different client numbers. We compare the model accuracy of our method with benchmark methods for the first 5 clients. In federated learning-based approaches, more client participation can improve the model performance. We can observe that the proposed method always achieves the highest accuracy for different client numbers. Source data are provided as a Source Data file.



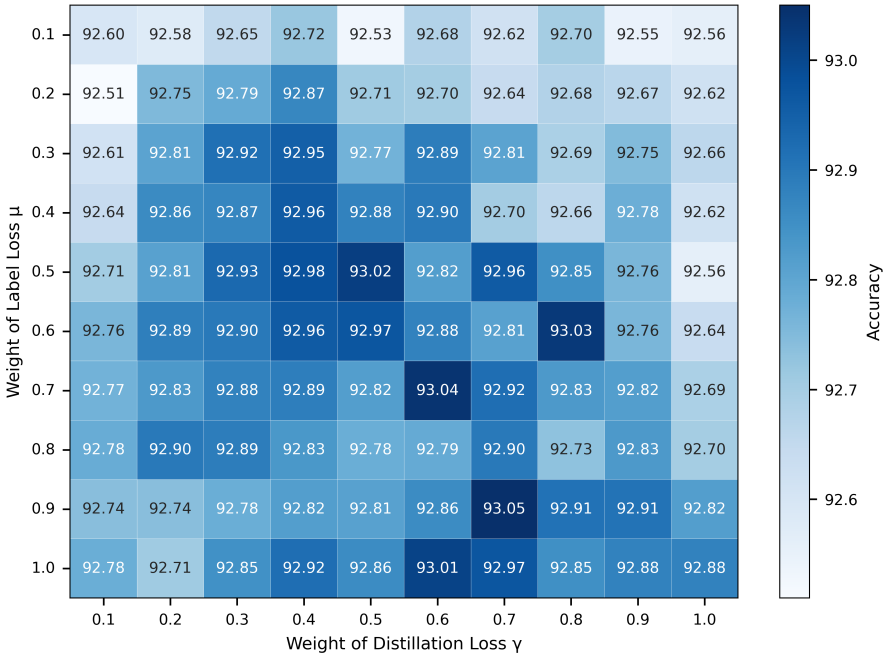
Supplementary Fig. 3: Comparison of forecasting accuracy versus training time for six on-device feasible methods. The federated learning-based method performs 30 rounds of local fine-tuning. We can see that the proposed method achieves the best performance while significantly reducing the training time compared to vanilla federated split learning methods. Source data are provided as a Source Data file.



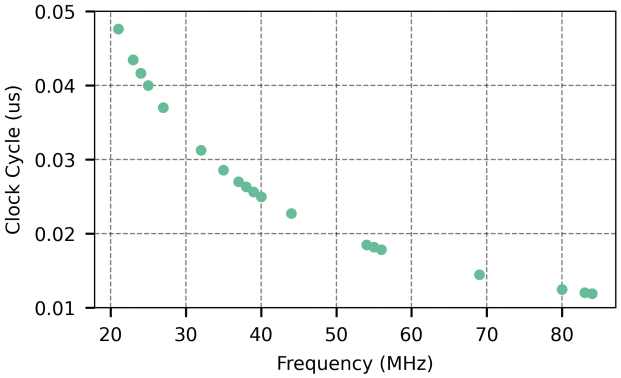
Supplementary Fig. 4: Comparison of forecasting accuracy when splitting at different hidden layers. The experimental results reveal that splitting at different hidden layers does not significantly affect the forecasting accuracy of the model on both datasets. Source data are provided as a Source Data file.



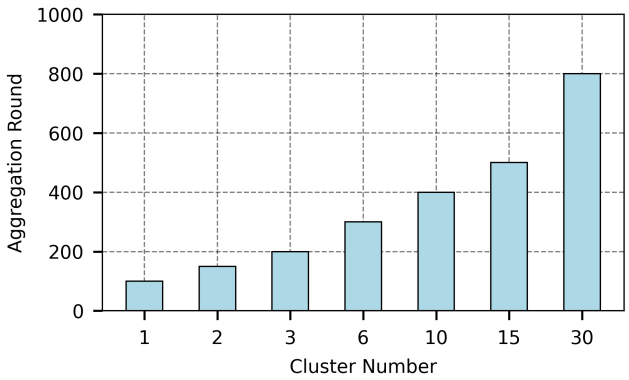
Supplementary Fig. 5: Effectiveness of the efficiency-optimal model splitting strategy considering deeper neural network with 7 hidden layers. Each hidden layer is considered a split layer. The split layers for best efficiency are annotated. The hidden layers contained in the feature extractor, feature processor, and regressor after the optimal split are indicated with different colours. Total training time under four distinct hardware configurations when choosing different split layers is provided. The stacked histograms represent the measured time for communication, forward propagation of the edge server and smart meter, and parallel backward propagation, arranged from bottom to top. Source data are provided as a Source Data file.



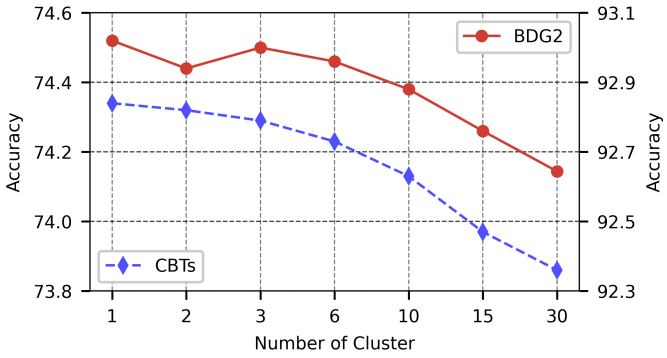
Supplementary Fig. 6: Comparison of forecasting accuracy for choosing different loss weights on dataset BDG2. Columns: weight of distillation loss. Rows: weight of label loss. We find that models trained with too small label loss weights perform poorly. The model generally performs better when the two weights are closer together. Source data are provided as a Source Data file.



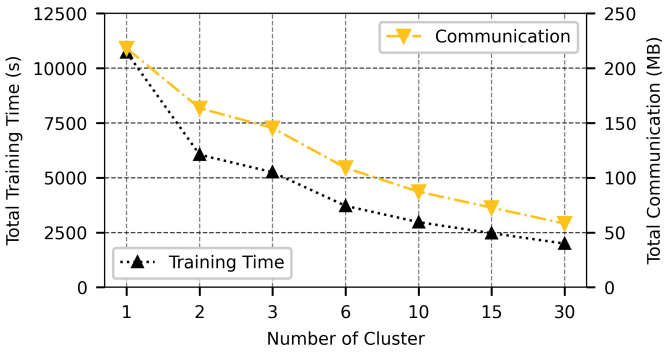
Supplementary Fig. 7: Heterogeneous hardware configurations for 30 smart meters. To simulate the device heterogeneity in a real smart grid, 30 smart meters are randomly set to different operating frequencies between 21MHz and 84MHz, with an average frequency of 42MHz. The operational frequency settings of smart meters in our experiments range from 1/2 to 1/8 of the maximum values, which is an engineering experience value derived from the perspective of task occupancy rate. In this setup, the computational power of the fastest smart meter is about four times the computational power of the slowest one. Note that the clock cycle is inversely proportional to the operating frequency. Source data are provided as a Source Data file.



Supplementary Fig. 8: Aggregation round selected for the semi-asynchronous aggregation method. In each edge-cloud aggregation, only one cluster uploads the model gradient to update the global model in each round. We can observe that as the number of clusters increases, the deviation of individual cluster gradients makes the training rounds longer for convergence. Note that clusters with shorter training times upload model gradients more frequently than clusters with longer training times at a given time. Source data are provided as a Source Data file.

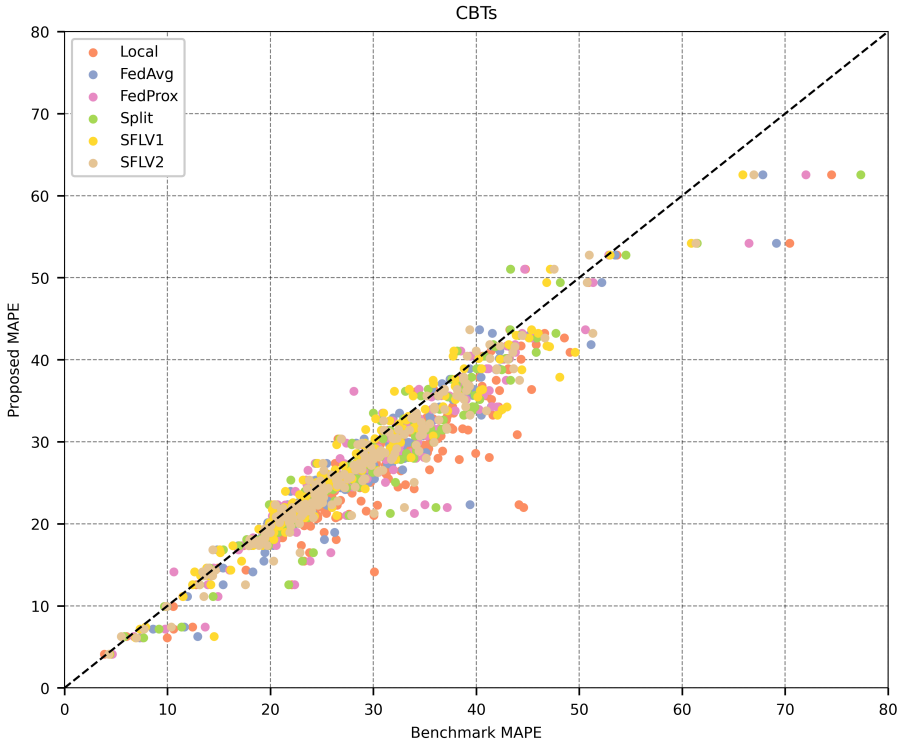


(a)

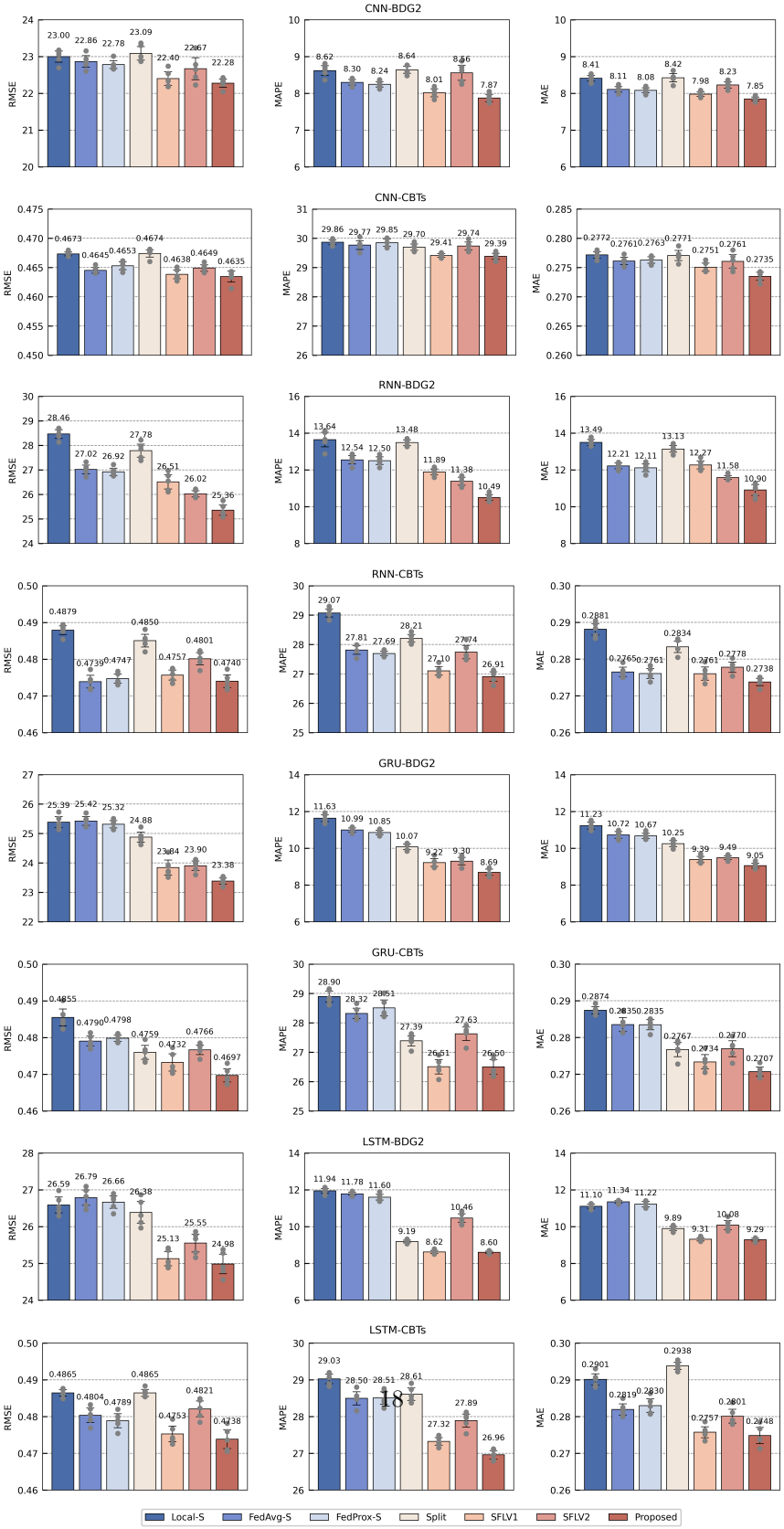


(b)

Supplementary Fig. 9: Impact of the cluster number on the model performance. (a) Schematic diagram of edge energy management for buildings/homes with flexible energy resources. (b) Comparison of training time and communication versus cluster number. Similarly, the training time and communication overhead decrease at a faster rate as the number of clusters increases. It shows that our method can dramatically enhance training efficiency without sacrificing precision. Source data are provided as a Source Data file.



Supplementary Fig. 10: Accuracy comparison of benchmark and proposed methods in a large-scale scenario. The performance of the benchmark models is indicated by the x-axis, and the y-axis indicates that of the proposed model. Points below the dashed line indicate households where the proposed model performs better than the baseline model. The fact that most of the points are below the dashed line implies that the proposed model can perform well on most households and achieve significant improvement on a few of them. Source data are provided as a Source Data file.



Supplementary Fig. 11: Performance evaluation of the proposed method with different neural networks as the backbone. We compare the accuracy of our method with other device-friendly methods with a CNN, RNN, GRU, or LSTM as the backbone. The mean accuracy with 95% confidence intervals is presented with 5 independent experiments. Source data are provided as a Source Data file.

Supplementary Tables

Supplementary Table 1: Number of network parameters for various layers

Layer	$\lvert\mathbf{w}\rvert$	$\lvert\mathbf{b}\rvert$
Fully connected layer	$M{\cdot}N$	N
Convolutional layer	$F{\cdot}C{\cdot}K$	F
Recurrent layer	$(M{+}N){\cdot}N$	N
LSTM layer	$4{\cdot}(M{+}N){\cdot}N$	$4{\cdot}N$

M: number of neurons in the previous layer.
N: number of neurons in the current layer.
F: number of filters in the previous layer.
N: number of filters in the current layer.
K: kernel size.

Supplementary Table 2: Number of intermediate parameters for various layers

Layer	$\lvert\mathbf{a}\rvert$	$\lvert\frac{\partial l}{\partial \mathbf{w}}\rvert$	$\lvert\frac{\partial l}{\partial \mathbf{b}}\rvert$	$\lvert\frac{\partial l}{\partial \mathbf{a}}\rvert$
Fully connected layer	N	$M{\cdot}N$	N	N
Convolutional layer	$F{\cdot}(M\text{-}K{+}1)$	$F{\cdot}C{\cdot}K$	F	$C{\cdot}N$
Recurrent layer	$S{\cdot}N$	$(M{+}N){\cdot}N$	N	$S{\cdot}N$
LSTM layer	$S{\cdot}N$	$4{\cdot}(M{+}N){\cdot}N$	$4{\cdot}N$	$S{\cdot}N$
Activation layer	M	-	-	M
Pooling layer	F	-	-	$F{\cdot}M$
Flatten layer	$F{\cdot}M$	-	-	$F{\cdot}M$

S: length of input series.

Supplementary Table 3: Performance of different methods on BDG2 and CBTs in terms of accuracy, memory, training time, and communication overhead per round when considering deeper 7 hidden layers.

Method	BDG2			CBTs			Memory (KB)	Training Time (s)	Communication (KB)
	RMSE	MAPE	MAE	RMSE	MAPE	MAE			
Local-M	22.75	7.68	8.03	0.4678	26.68	0.2721	5178.25 (1.0x)	4335.86 (1.01x)	-
FedAvg-M	<u>22.33</u>	**6.84**	7.44	**0.4617**	26.28	<u>0.2638</u>	5178.25 (1.0x)	4387.33 (1.0x)	761.5 (1.0x)
FedProx-M	22.42	7.01	7.52	<u>0.4620</u>	<u>26.17</u>	0.2641	5178.25 (1.0x)	4387.33 (1.0x)	761.5 (1.0x)
Split	22.93	7.89	8.14	0.4694	26.74	0.2713	103.75 (49.9x)	281.86 (15.55x)	91.25 (8.34x)
SFLV1	22.56	7.03	7.57	0.4963	26.43	0.2650	103.75 (49.9x)	282.84 (15.51x)	96.75 (7.87x)
SFLV2	22.75	7.18	7.66	0.4647	26.39	0.2663	103.75 (49.9x)	282.84 (15.51x)	96.75 (7.87x)
Proposed	**22.22**	<u>6.86</u>	**7.38**	0.4626	**26.12**	**0.2637**	115.07 (45x)	214.68 (20.43x)	74.43 (10.23x)

[1] The best-performing and the second-best-performing methods are bolded and underlined, respectively.
[2] -M indicates the model has multiple hidden layers.

Supplementary Table 4: Performance evaluation of different methods on BDG2 and CBTs under passive membership inference attack.

Method	Attack Accuracy	
	BDG2	CBTs
FedAvg-S	0.6738	0.4804
Split	0.8387	0.4628
SFLV1	0.5783	0.3894
SFLV2	0.6012	0.4152
Proposed	**0.532**	**0.3849**

Supplementary Proofs

Supplementary Proof 1: Efficiency-optimal split ratio

Considering the first case, we have $\alpha \geq (\frac{P_{es}}{KP_{sm}} + 1)^{-1}$ which is equivalent to

$$\frac{\alpha\,(1-\beta)\,n\,|D|\,|\mathbf{w}|}{P_{sm}} \geq \frac{(1-\alpha)\,(1-\beta)\,n\,|D|\,|\mathbf{w}|\,K}{P_{es}} \tag{9}$$

Hence, the training time can be rewritten as

$$T = \frac{3s\,|D| + 2\alpha\,|\mathbf{w}|}{R} + \frac{\alpha n\,|D|\,|\mathbf{w}|}{P_{sm}} + \frac{(1-\alpha)n\beta\,|D|\,|\mathbf{w}|\,K}{P_{es}} \tag{10}$$

Here when $P_{es} \geq \beta K(\frac{2}{nR|D|} + \frac{1}{P_{sm}})^{-1}$, (10) is an increasing function of α and the optimal ratio minimizing the training time is $\alpha^* = (\frac{P_{es}}{KP_{sm}} + 1)^{-1}$ provided that α^* is not larger than the upper bound α_{upper}. Otherwise (10) is a decreasing function, i.e., the optimal ratio is the upper bound α_{upper}.

Then, considering the second case, we have $\alpha \leq (\frac{P_{es}}{KP_{sm}} + 1)^{-1}$, which is equivalent to

$$\frac{\alpha\,(1-\beta)\,n\,|D|\,|\mathbf{w}|}{P_{es}} \leq \frac{(1-\alpha)\,(1-\beta)\,n\,|D|\,|\mathbf{w}|\,K}{P_{sm}} \tag{11}$$

Hence, the training time can be rewritten as

$$T = \frac{3s\,|D| + 2\alpha\,|\mathbf{w}|}{R} + \frac{\alpha\beta n\,|D|\,|\mathbf{w}|}{P_{sm}} + \frac{(1-\alpha)\,n\,|D|\,|\mathbf{w}|\,K}{P_{es}} \tag{12}$$

Here when $P_{es} \leq K(\frac{1}{nR|D|} + \frac{\beta}{P_{sm}})^{-1}$, (12) is a decreasing function of α and the optimal ratio minimizing the training time is $\alpha^* = (\frac{P_{es}}{KP_{sm}} + 1)^{-1}$ provided that α^* is not smaller than the lower bound α_{lower}. Otherwise (12) is an increasing function, i.e., the optimal ratio is the lower bound α_{lower}. Note that β is between 0 and 1, so the upper bound is larger than the lower one. Hence, we complete the proof of the optimal ratio.

Supplementary Proof 2: Convergence of auxiliary model

Under Assumption 1, we can write

$$\mathcal{L}_c(\mathbf{w}_c^{t+1}) - \mathcal{L}_c(\mathbf{w}_c^t) \leq \nabla \mathcal{L}_c(\mathbf{w}_c^t)^T(\mathbf{w}_c^{t+1} - \mathbf{w}_c^t) + \frac{L}{2}\left\|\mathbf{w}_c^{t+1} - \mathbf{w}_c^t\right\|^2. \tag{13}$$

Note that we have

$$\mathbf{w}_c^{t+1} = \mathbf{w}_c^t - \tau_i \eta_t \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \tag{14}$$

Substituting $\mathbf{w}_c^{t+1} - \mathbf{w}_c^t$, we have

$$\begin{aligned} \mathcal{L}_c(\mathbf{w}_c^{t+1}) - \mathcal{L}_c(\mathbf{w}_c^t) \leq & -\tau_i \eta_t \nabla \mathcal{L}_c(\mathbf{w}_c^t)^T \left(\frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right) \\ & + \frac{L}{2} \tau_i^2 \eta_t^2 \left\| \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right\|^2. \end{aligned} \tag{15}$$

Taking expectation over (15) as follows

$$\begin{aligned} \mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^{t+1})\right] - \mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^t)\right] \leq & -\tau_i \eta_t \underbrace{\mathbb{E}\left[\nabla \mathcal{L}_c(\mathbf{w}_c^t)^T \left(\frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right) \right]}_{A_1} \\ & + \frac{L}{2} \tau_i^2 \eta_t^2 \underbrace{\mathbb{E}\left[\left\| \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right\|^2 \right]}_{A_2}. \end{aligned} \tag{16}$$

In the following, we will bound A_1 and A_2, respectively. The equivalent form of A_1 is

$$\begin{aligned} A_1 &= \mathbb{E}\left[\nabla \mathcal{L}_c(\mathbf{w}_c^t)^T \left(\frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_c^t) \right) \right] \\ &= \frac{1}{2}\mathbb{E}\left[\left\| \nabla \mathcal{L}_c(\mathbf{w}_c^t) \right\|^2 \right] + \frac{1}{2}\mathbb{E}\left[\left\| \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right\|^2 \right] \\ &\quad - \frac{1}{2} \underbrace{\mathbb{E}\left[\left\| \nabla \mathcal{L}_c(\mathbf{w}_c^t) - \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t) \right\|^2 \right]}_{B}. \end{aligned} \tag{17}$$

Since the sampled gradient of each cluster is an unbiased estimator of the full gradient, i.e., $\nabla \mathcal{L}_c(\mathbf{w}_c^t) = \mathbb{E}\left[\frac{1}{K_i}\sum_{k\in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t)\right]$, so we have

$$\begin{aligned} B &= \mathbb{E}\left[\left\|\nabla \mathcal{L}_c(\mathbf{w}_c^t) - \frac{1}{K_i}\sum_{k\in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t)\right\|^2\right] \\ &= \mathbb{E}\left[\left\|\mathbb{E}\left[\frac{1}{K_i}\sum_{k\in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t)\right] - \frac{1}{K_i}\sum_{k\in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t)\right\|^2\right] \\ &\leq \mathbb{E}\left[\left\|\frac{1}{K_i}\sum_{k\in A_i} \nabla \mathcal{L}_{c,k}(\mathbf{w}_{c,k}^t)\right\|^2\right] - \mathbb{E}\left[\left\|\nabla \mathcal{L}_c(\mathbf{w}_c^t)\right\|^2\right] \end{aligned} \tag{18}$$

Substituting (18) into (17), we have

$$A_1 = \mathbb{E}\left[\left\|\nabla \mathcal{L}_c\left(\mathbf{w}_c^t\right)\right\|^2\right]. \tag{19}$$

Then we apply Cauchy-Schwarz inequality and Assumption 2 to find the lower bound of A_2:

$$\begin{aligned} A_2 &= \mathbb{E}\left[\left\|\frac{1}{K}\sum_{k\in A_t} \nabla \mathcal{L}_{c,k}\left(\mathbf{w}_{c,k}^t\right)\right\|^2\right] \\ &\leq \frac{1}{K_i}\sum_{k\in A_i} \mathbb{E}\left[\left\|\nabla \mathcal{L}_{c,k}\left(\mathbf{w}_{c,k}^t\right)\right\|^2\right] \\ &\leq \frac{1}{K_i}\sum_{k\in A_i} \frac{1}{|D_k|}\sum_{x\in D_k} \mathbb{E}\left[\left\|\nabla \ell_{c,k}(\mathbf{w}_{c,k}, x_k)\right\|^2\right] \\ &\leq G_1. \end{aligned} \tag{20}$$

Substituting (19) and (20) into (16), it follows that

$$\mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^{t+1})\right] \leq \mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^t)\right] - \tau_i\eta_t\mathbb{E}\left[\left\|\nabla \mathcal{L}_c(\mathbf{w}_c^t)\right\|^2\right] + G_1\frac{L}{2}\tau_i^2\eta_t^2. \tag{21}$$

Now by summing up for all global rounds $t = 1, ...T$, we have

$$\mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^T)\right] \leq \mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^0)\right] - \tau_i\sum_{t=1}^{T}\eta_t\mathbb{E}\left[\left\|\nabla \mathcal{L}_c(\mathbf{w}_c^t)\right\|^2\right] + G_1\frac{L}{2}\tau_i^2\sum_{t=1}^{T}\eta_t^2. \tag{22}$$

Finally due to $\mathcal{L}_c(\mathbf{w}_c^*) \leq \mathbb{E}\left[\mathcal{L}_c(\mathbf{w}_c^T)\right]$, we have

$$\sum_{t=1}^{T}\eta_t\mathbb{E}\left[\left\|\nabla \mathcal{L}_c(\mathbf{w}_c^t)\right\|^2\right] \leq \frac{1}{\tau_i}\left[\mathcal{L}_c(\mathbf{w}_c^0) - \mathcal{L}_c(\mathbf{w}_c^*)\right] + G_1\frac{L}{2}\tau_i\sum_{t=1}^{T}\eta_t^2, \tag{23}$$

where τ_i can be regarded as a constant due to the equal number of smart meters in each cluster. Hence, we complete the convergence proof for the auxiliary model.

Supplementary Proof 3: Convergence of main model

Under Assumption 1, We obtain the same form as the auxiliary model

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^{t+1})\right] - \mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^{t})\right] \leq & \tau_i\eta_t \underbrace{\mathbb{E}\left[-\nabla\mathcal{L}_s(\mathbf{w}_s^t)^T\left(\frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t)\right)\right]}_{C_1} \\ & + \frac{L}{2}\tau_i^2\eta_t^2 \underbrace{\mathbb{E}\left[\left\|\frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t)\right\|^2\right]}_{C_2}.\end{aligned} \tag{24}$$

In the following, we will bound C_1 and C_2, respectively. Applying the Cauchy-Schwarz inequality, we have

$$\begin{aligned}C_1 &\leq \mathbb{E}\left[\nabla\left\|\mathcal{L}_s(\mathbf{w}_s^t)\right\|^2 - \nabla\mathcal{L}_s(\mathbf{w}_s^t)^T\left(\frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t)\right)\right] \\ &= \mathbb{E}\left[\nabla\mathcal{L}_s(\mathbf{w}_s^t)^T\left(\nabla\mathcal{L}_s(\mathbf{w}_s^t) - \frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t)\right)\right] \\ &\leq \underbrace{\mathbb{E}\Big[\left\|\nabla\mathcal{L}_s(\mathbf{w}_s^t)\right\|\Big]}_{D_1}\cdot\underbrace{\mathbb{E}\left[\left\|\nabla\mathcal{L}_s(\mathbf{w}_s^t) - \frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t)\right\|\right]}_{D_2}\end{aligned} \tag{25}$$

We first find a lower bound of D_1 as

$$\begin{aligned}D_1 &= \mathbb{E}\Big[\left\|\nabla\mathcal{L}_s(\mathbf{w}_s^t)\right\|\Big] \\ &\leq \sqrt{G_2}\end{aligned} \tag{26}$$

We apply Fubini's theorem to find a lower bound of D_2 as

$$\begin{aligned}D_2 &= \mathbb{E}\left[\left\|\nabla\mathcal{L}_s(\mathbf{w}_s^t) - \frac{1}{K_i}\sum_{k\in A_i}\nabla\mathcal{L}_{s,k}(\mathbf{w}_s^t)\right\|\right] \\ &\leq \frac{1}{K_i}\sum_{k\in A_i}\mathbb{E}\left[\left\|\frac{1}{|D_k|}\sum_{x\in D_k}\nabla\ell_{s,k}(\mathbf{w}_{s,k}^t, h_{e,k}^*(x)) - \nabla\ell_{s,k}\left(\mathbf{w}_{s,k}^t, h_{e,k}^t(x)\right)\right\|\right] \\ &\leq \frac{1}{K_i}\sum_{k\in A_i}\mathbb{E}\left[\int\left\|\nabla\ell_{s,k}(\mathbf{w}_{s,k}^t, h)\right\|\left\|p_{c,k}^t(h) - p_{c,k}^*(h)\right\|dh\right] \\ &\leq \sqrt{G_2}\frac{1}{K_i}\sum_{k\in A_i}\sqrt{d_{c,k}^t}.\end{aligned} \tag{27}$$

By combining (25), (26) and (27), we have

$$C_1 \leq G_2 \tau_i \eta_t \frac{1}{K_i} \sum_{k \in A_i} \sqrt{d_{c,k}^t} - \tau_i \eta_t \mathbb{E}\left[\nabla \left\| \mathcal{L}_s(\mathbf{w}_s^t) \right\|^2\right]. \tag{28}$$

With the same derivation process for the auxiliary model, we find a lower bound of C_2 as

$$\begin{aligned} C_2 &= \mathbb{E}\left[\left\| \frac{1}{K_i} \sum_{k \in A_i} \nabla \mathcal{L}_{s,k}(\mathbf{w}_{s,k}^t) \right\|^2\right] \\ &\leq G_2. \end{aligned} \tag{29}$$

Substituting (28) and (29) into (24), it follows that

$$\mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^{t+1})\right] \leq \mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^t)\right] - \tau_i \eta_t \mathbb{E} \left\| \nabla \mathcal{L}_s(\mathbf{w}_s^t) \right\|^2 + G_2 \left(\tau_i \eta_t \frac{1}{K_i} \sum_{k \in A_i} \sqrt{d_{c,k}^t} + \frac{L}{2} \tau_i^2 \eta_t^2 \right). \tag{30}$$

Now by summing up for all global rounds t = 1, ...T, we have

$$\begin{aligned} \mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^T)\right] \leq \mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^0)\right] &- \tau_i \sum_{t=1}^{T} \eta_t \mathbb{E}\left[\left\| \nabla \mathcal{L}_s(\mathbf{w}_s^t) \right\|^2\right] \\ &+ G_2 \left(\tau_i \sum_{t=1}^{T} \eta_t \frac{1}{K_i} \sum_{k \in A_i} \sqrt{d_{c,k}^t} + \frac{L}{2} \tau_i^2 \sum_{t=1}^{T} \eta_t^2 \right). \end{aligned} \tag{31}$$

Finally due to $\mathcal{L}_s(\mathbf{w}_s^*) \leq \mathbb{E}\left[\mathcal{L}_s(\mathbf{w}_s^T)\right]$, we have

$$\sum_{t=1}^{T} \eta_t \mathbb{E}\left[\left\| \nabla \mathcal{L}_s(\mathbf{w}_s^t) \right\|^2\right] \leq \frac{1}{\tau_i} \left[\mathcal{L}_s(\mathbf{w}_s^0) - \mathcal{L}_s(\mathbf{w}_s^*)\right] + G_2 \left(\sum_{t=1}^{T} \eta_t \frac{1}{K_i} \sum_{k \in A_i} \sqrt{d_{c,k}^t} + \frac{L}{2} \tau_i \sum_{t=1}^{T} \eta_t^2 \right), \tag{32}$$

Since Supplementary Proof. 2 has proved that the auxiliary model is convergent, we have $\sum_{t=1}^{T} d_{c,k}^t < \infty$. Hence, we complete the convergence proof for the main model.