# Introducing Edge Intelligence into Smart Meters

**Yi Wang**

yiwang@eee.hku.hk

**Assistant Professor, EEE**

**The University of Hong Kong**

June 2024

# About Myself

## Appointment

2021.9- Assistant Professor, The University of Hong Kong

2019.2-2021.8 Postdoc, ETH Zurich (Prof. Gabriela Hug)

## Education

2014.9-2019.1 Ph.D., Tsinghua University (Prof. Chongqing Kang)

2017.3-2018.4 Visiting Student, University of Washington (Prof. Daniel Kirschen)

2010.9-2014.6 B.S., Huazhong University of Science and Technology

## Research Interests

Data analytics for smart energy

Cyber-physical power and energy systems

Multi-energy systems integration

# Research Group: EDL@HKU

- Energy Digitalization Laboratory at The University of Hong Kong (EDL@HKU) focuses on the digitalization of power and energy systems with an emphasis on the distribution and consumer side, including **data analytics, data privacy, cyber-physical-social systems, Internet-of-things**, etc. The overall goal is to make the distribution systems more adaptive to accommodate the high penetration of renewable energy toward a decarbonized future.

In addition to publishing research papers, we develop/provide:

- Software
- Hardware
- Technical reports
- Policy recommendations

- **This presentation is based on the following on-going work:**


[1] Yehui Li, Dalin Qin, Vincent H. Poor, and Yi Wang, “Introducing Edge Intelligence into Smart Meters via Federated Split Learning,” Nature Communications, under review.

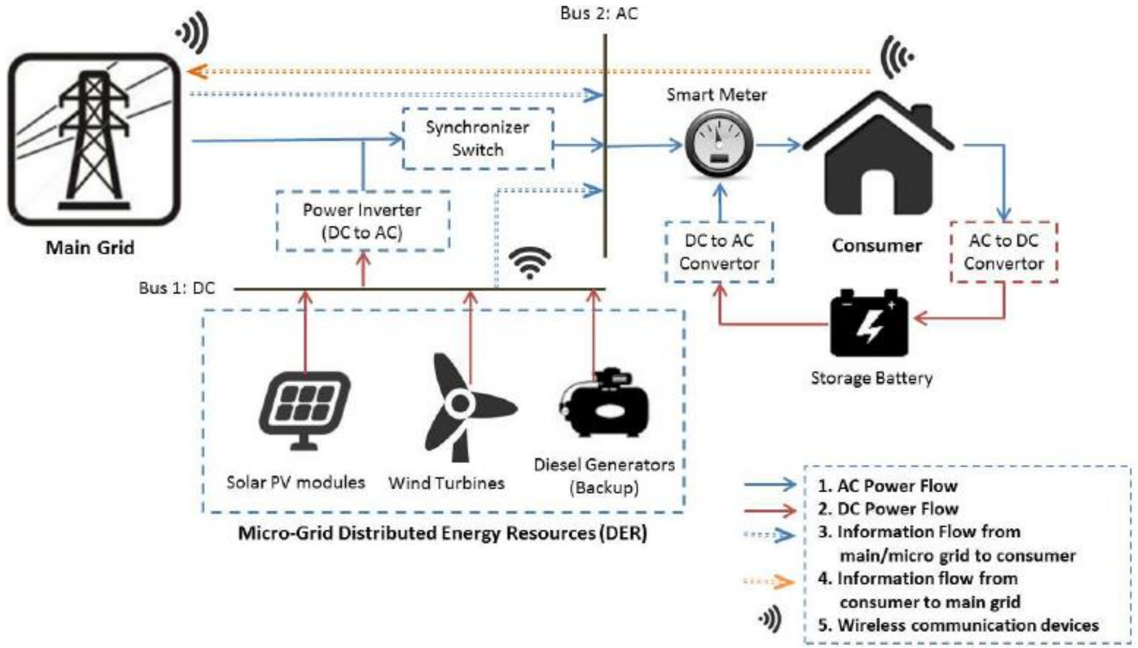
[2] Yehui Li, Ruiyang Yao, Dalin Qin, and Yi Wang, “Lightweight Federated Learning for On-Device Non-Intrusive Load Monitoring,” IEEE Transactions on Smart Grid, under review.

[3] Yehui Li, Xianhao Chen, and Yi Wang, “EdgeHEM: Sparse Federated Reinforcement Learning for Home Energy Management at the Edge,” IEEE Transactions on Smart Grid, under review.

# Contents

# Introduction

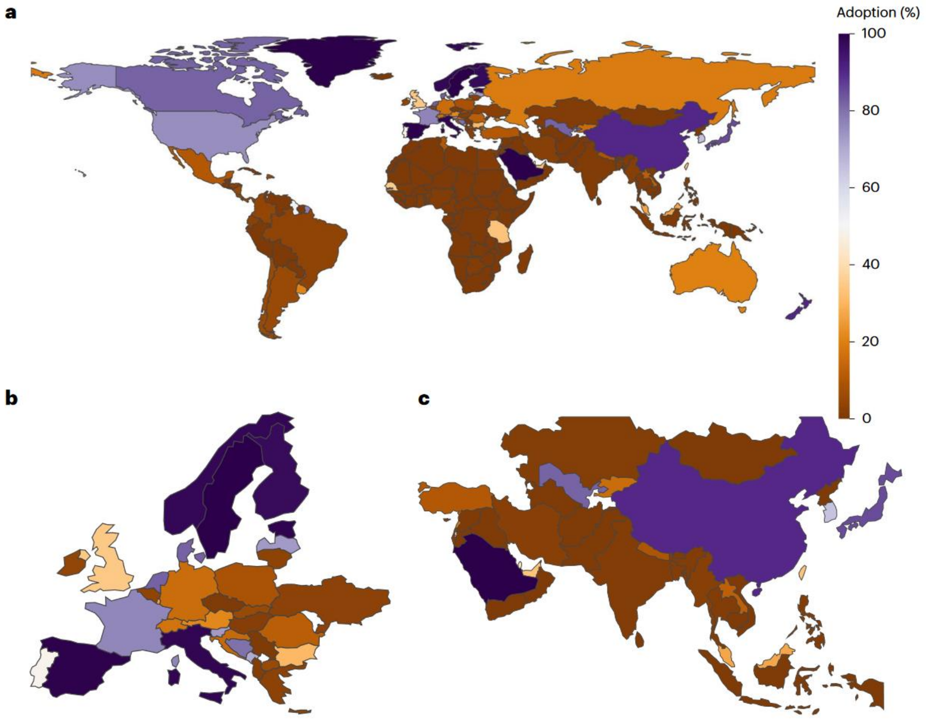
- Electric power systems account for over **40%** of global carbon dioxide emissions. Accommodating high penetration of renewable energy is an essential way to decarbonize power systems and thus alleviate climate change.
- Harnessing **demand-side flexibility** is a cost-effective strategy to promote renewable energy accommodation

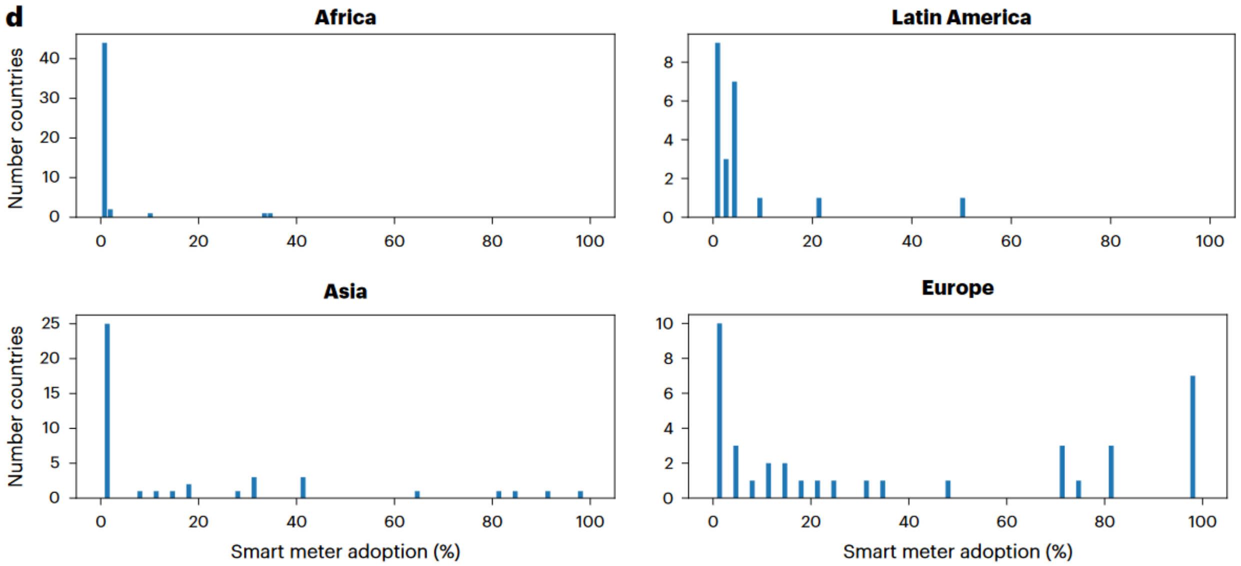



O'Shaughnessy E, Shah M, Parra D, et al. The demand-side resource opportunity for deep grid decarbonization[J]. Joule, 2022, 6(5): 972-983.

# Introduction

- The number of global smart meters is expected to exceed **1.2 billion** by the end of 2024, and the global penetration of smart meters will rise to nearly **59%** by 2028.
- The ubiquitous smart meters become the central feature of future smart grids by enabling the collection of massive fine-grained consumption data to **support demand-side flexibility**.




Aryandoust A, Patt A, Pfenninger S. Enhanced spatio-temporal electric load forecasts using less data with active deep learning[J]. Nature Machine Intelligence, 2022, 4(11): 977-991.

# Introduction

- However, the current smart meters are **still not smart enough**. They are incapable of conducting on-device intelligent data analytics but can only transmit immense collected data to the data management system, which results in **privacy violations**, **heavy transmission burdens**, and **low efficiency in demand-side management**.

***Enabling smart meter intelligence***

- Enabling on-device intelligence for existing ubiquitous smart meters without additional investment in computational facilities **is the most economical way** to facilitate consumers managing flexible resources more autonomously and efficiently.
- Enabling smart meter intelligence can reduce the need for local data uploading, which may **alleviate privacy concerns** and improve consumers' willingness to smart meter adoption.

# Introduction

- Massive collected load data can be locally transformed into knowledge, provide deeper insights into the **present**, better understandings of the **future**, and practical advice on possible **decisions** for the smart grid.
- However, existing data analysis methods are not applicable to smart meters due to the limitation of **data availability** and **hardware resources**:
  - Smart meter data involves consumers' privacy, which causes the data barrier hindering the utilization of distributed big data.
  - Smart meters have insufficient memory, computation, and communication resources to support the complicated model training.

Data privacy concerns hinder the utilization of big data

Resource constraints hinder the training of large models

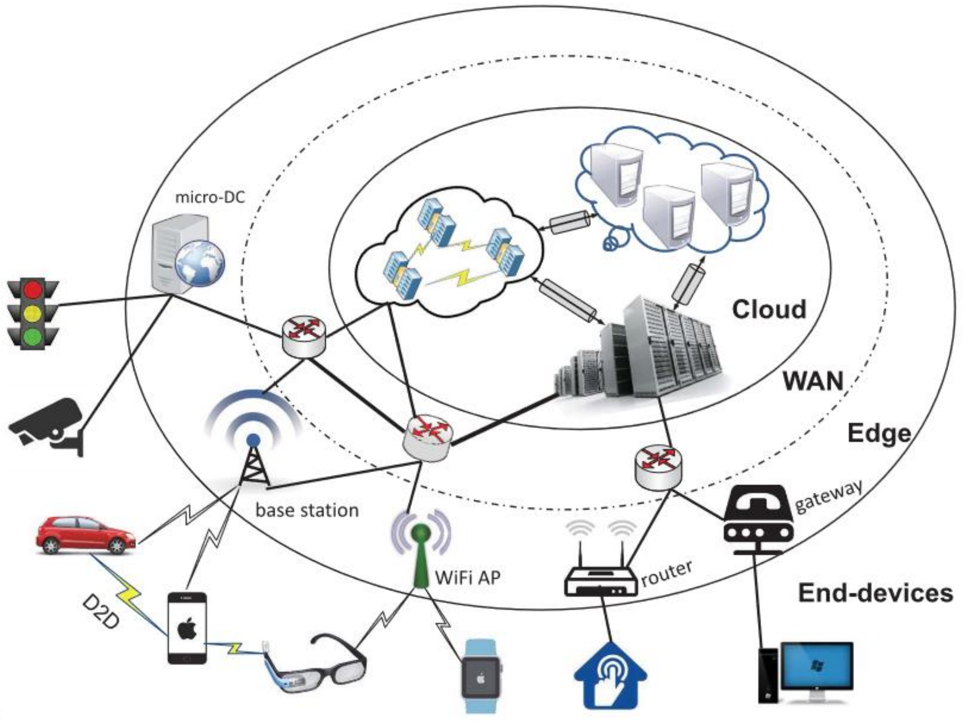
Véliz C, Grunewald P. Protecting data privacy is key to a smart energy future[J]. Nature Energy, 2018, 3(9): 702-704.

# Introduction

- Recently, researchers have focused on harnessing the potential of edge big data and computational resources by pushing artificial intelligence toward end devices, giving rise to the concept of "**edge intelligence**" (EI).

- Considering the data privacy concerns as well as hardware constraints of smart meters in terms of memory, computation, and communication capacity, **achieving EI on smart meters requires a privacy-preserving framework with high efficiency**.

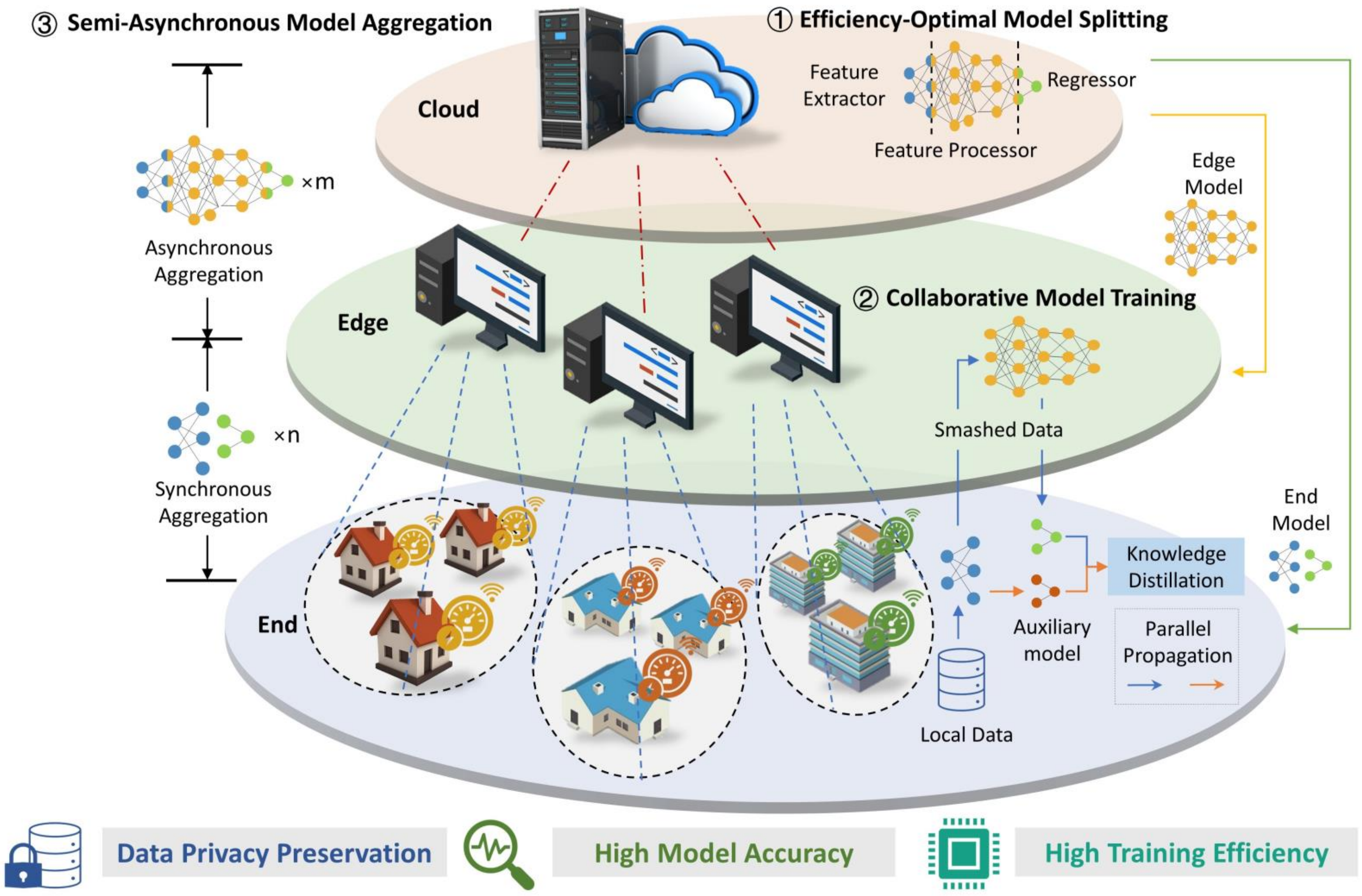
- However, a unified framework to achieve EI on smart meters is still lacking.




Zhou Z, Chen X, Li E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

# Contents

# Framework




**Overview of the end-edge-cloud framework**

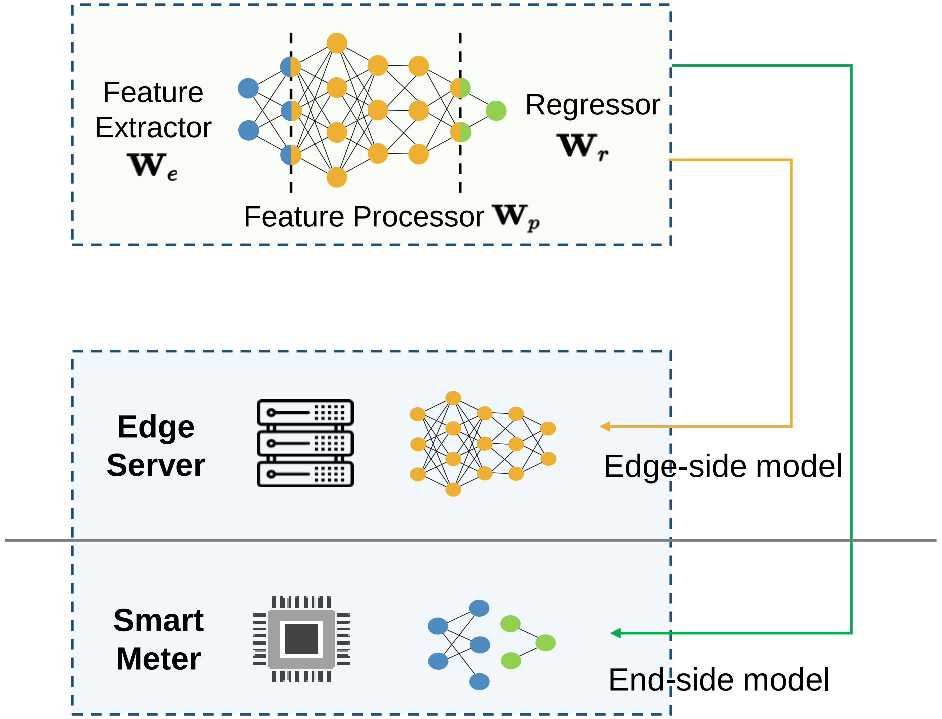
- The framework enables collaborative training of the model deployed on different entities with distributed data in a privacy-enhancing way.

① **Model splitting**: cloud server splits the large model with optimal ratio and assigns a small portion to smart meters and a larger portion to the edge servers

② **Model training**: multiple smart meters collaborate with edge servers to train the complete model in a efficient way

③ **Model aggregation**: the trained models are hierarchically aggregated by the edge servers and the cloud server to update the global model

# Optimal splitting




Define **split ratio** $\alpha$ as $\frac{|\mathbf{w}_e|+|\mathbf{w}_r|}{|\mathbf{w}|}$

Feature extractor and regressor involve **privacy raw data**

Feature processor requires **complex computation**

## Objective:

- we aim to find an optimal split ratio that minimizes the **training time** subject to the **memory constraints** of smart meters.

$$\min_{\alpha} T(\alpha)$$
$$\text{s.t. } M(\alpha) \le M_{sm}$$

## Modelling of memory footprint:

$$M(\alpha) = 32 \times \sum_{i=1}^{\lfloor \alpha L \rfloor} |B| \left(|\mathbf{w}_i| + 2|\mathbf{a}_i|\right) + 3|\mathbf{w}_i|$$

## Modelling of training time:

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

# Optimal splitting

**Solution for optimal split ratio:**

We can calculate the **upper bound** and **lower bound** as:

$$\alpha_{upper} = \inf\{\alpha : M(\alpha) \le M_{sm}\}$$

$$\alpha_{lower} = \frac{|\mathbf{w}_1| + |\mathbf{w}_L|}{|\mathbf{w}|}$$

The **optimal split ratio** can be formulated:

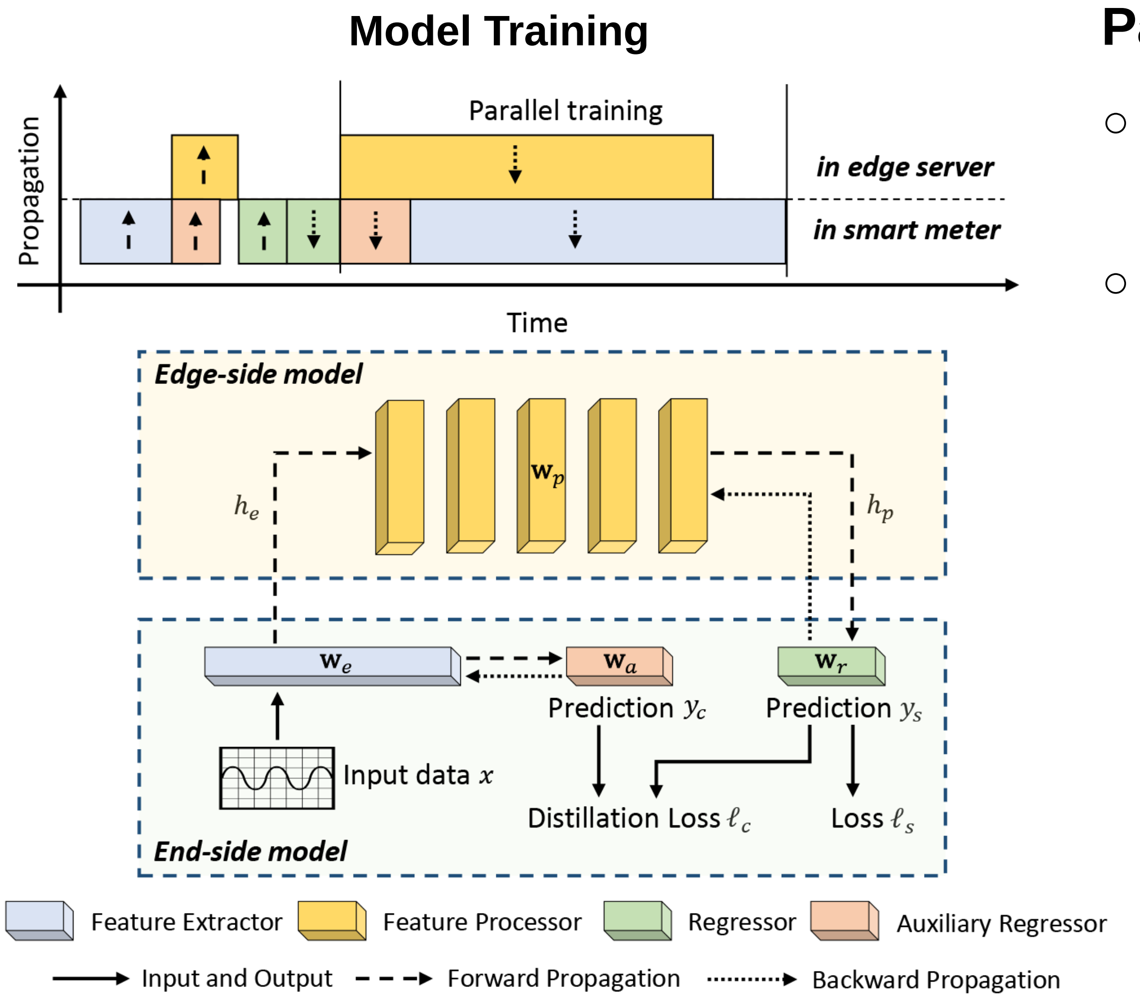
***Computational power of edge server***

***Computational power of smart meter***

***Communication rate***

$$\alpha^* = \begin{cases} \alpha_{\text{upper}} & \text{if } \boxed{P_{ES}} \ge K\left(\frac{1}{\boxed{R}|D|} + \frac{\beta}{\boxed{P_{ED}}}\right)^{-1} \\ \begin{cases} \alpha_{\text{upper}} & \text{if } \left(\frac{P_{ES}}{KP_{ED}} + 1\right)^{-1} \ge \alpha_{\text{upper}} \\ \left(\frac{P_{ES}}{KP_{ED}} + 1\right)^{-1} & \text{if } \alpha_{\text{upper}} \le \left(\frac{P_{ES}}{KP_{ED}} + 1\right)^{-1} \le \alpha_{\text{lower}} \\ \alpha_{\text{lower}} & \text{if } \left(\frac{P_{ES}}{KP_{ED}} + 1\right)^{-1} \le \alpha_{\text{lower}} \end{cases} & \text{if } \beta K\left(\frac{2}{R|D|} + \frac{1}{P_{ED}}\right)^{-1} \le P_{ES} \le K\left(\frac{1}{R|D|} + \frac{\beta}{P_{ED}}\right)^{-1} \\ \alpha_{\text{lower}} & \text{if } P_{ES} \le \beta K\left(\frac{2}{R|D|} + \frac{1}{P_{ED}}\right)^{-1} \end{cases}$$

# Collaborative training




## Parallelism:

- Inspired by distributed optimization, we add an **auxiliary network $\mathbf{w}_a$** as another regressor connected to **$\mathbf{w}_e$**.
- Here two trainable models are formed, namely:

$$\mathbf{w}_s = [\mathbf{w}_e, \mathbf{w}_p, \mathbf{w}_r] \qquad \mathbf{w}_c = [\mathbf{w}_e, \mathbf{w}_a].$$

Hence, **$\mathbf{w}_e$** and **$\mathbf{w}_p$** can update their parameters **in parallel** with different loss functions as:

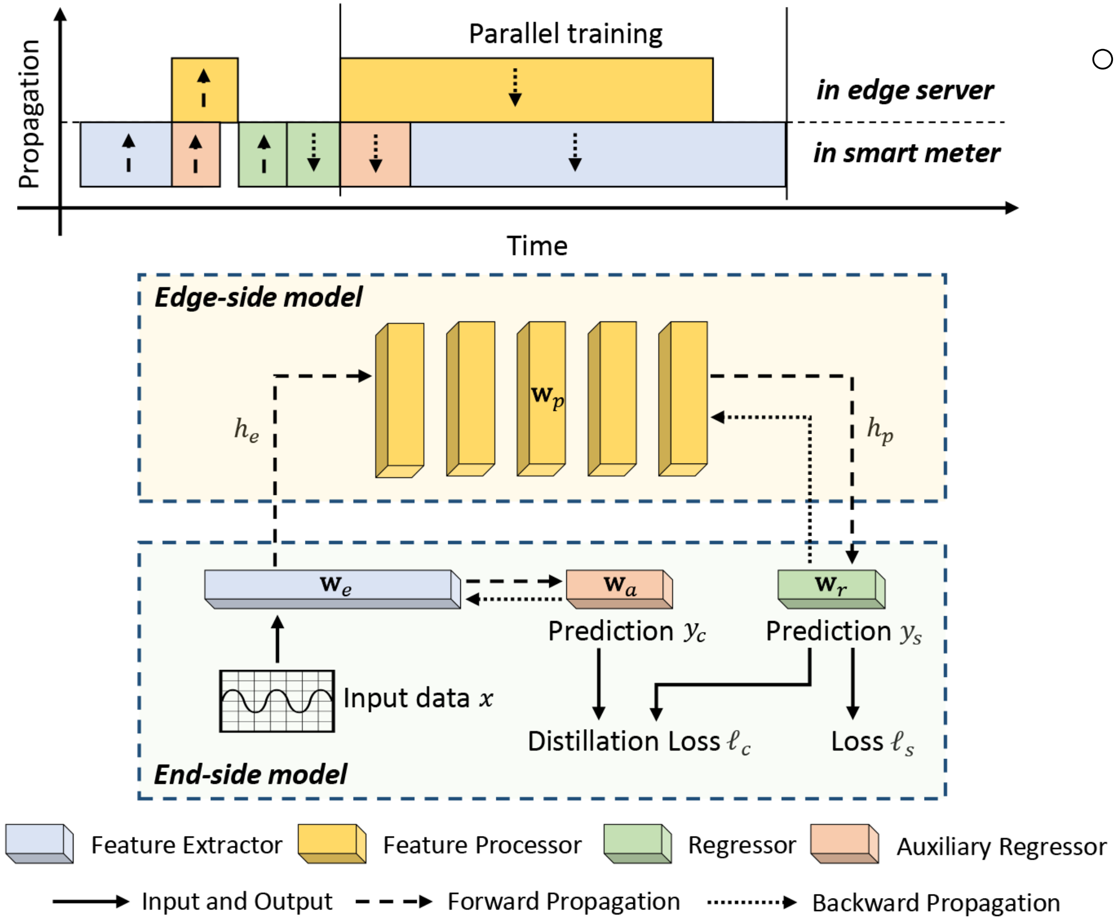
$$\min_{\mathbf{w}_r, \mathbf{w}_p} \mathcal{L}_s(\mathbf{w}_s) = \min_{\mathbf{w}_r, \mathbf{w}_p} \frac{1}{|D|} \sum_{x \in D} \ell_s(\mathbf{w}_s, x)$$

$$\min_{\mathbf{w}_a, \mathbf{w}_e} \mathcal{L}_c(\mathbf{w}_c) = \min_{\mathbf{w}_a, \mathbf{w}_e} \frac{1}{|D|} \sum_{x \in D} \ell_c(\mathbf{w}_c, x)$$

1. nearly half of the computational time is reduced
2. a quarter of the communication overhead is eliminated

# Collaborative training

### Model Training




## Knowledge distillation：

- $\mathbf{W}_e$ serve as a parameter in the equation (1), while it only be optimized based on the loss in the equation (2). This **lack of correlation** may affect the convergence accuracy.

$$\min_{\mathbf{w}_r,\mathbf{w}_p} \mathcal{L}_s(\boxed{\mathbf{w}_s}) = \min_{\mathbf{w}_r,\mathbf{w}_p} \frac{1}{|D|}\sum_{x\in D}\ell_s(\mathbf{w}_s,x) \qquad (1)$$

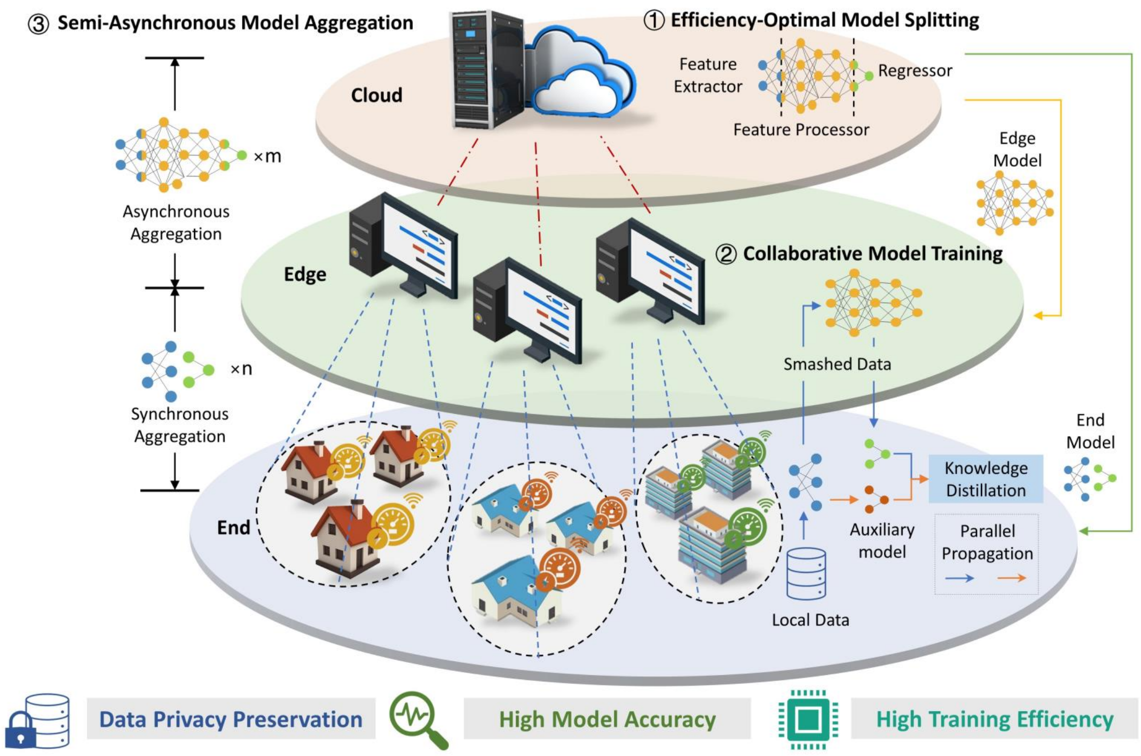
$$\min_{\mathbf{w}_a,\boxed{\mathbf{w}_e}} \mathcal{L}_c(\mathbf{w}_c) = \min_{\mathbf{w}_a,\mathbf{w}_e} \frac{1}{|D|}\sum_{x\in D}\ell_c(\mathbf{w}_c,x) \qquad (2)$$

Therefore, we design the loss function by incorporating **knowledge distillation** to introduce the convergence of (1) as an objective into the optimization of $\mathbf{W}_e$ .

$$\ell_s(\mathbf{w}_s,x) = \ell(y,y_s)$$
$$\ell_c(\mathbf{w}_c,x) = \mu\ell(y,y_c) + \underbrace{\gamma\ell(y_s,y_c)}_{\text{KnowledgeDistillation}}$$

# Semi-asynchronous aggregation

### Model aggregation




A two-stage approach can tackle the large-scale **heterogeneity** challenge of smart meters.

### End-edge synchronous aggregation:

- We adopt the hardware configuration-based **clustering algorithm** to designate smart meters with **similar training times** to the same edge server.
- Edge server aggregate intra-cluster smart meter's models **synchronously** as:

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

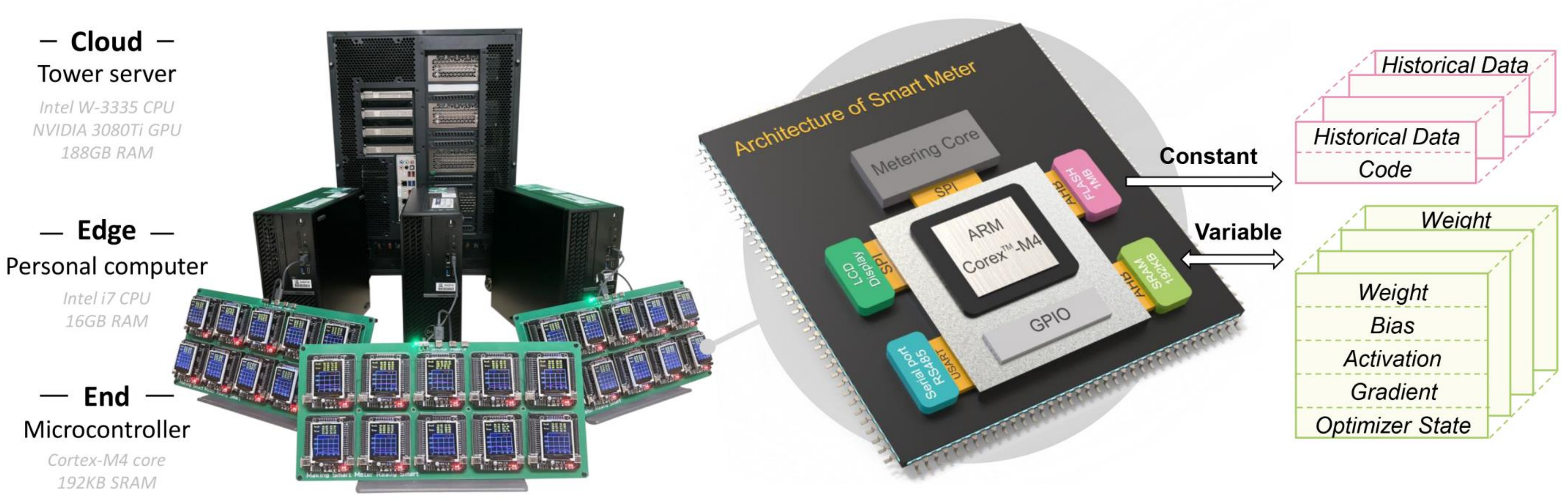
### Edge-cloud asynchronous aggregation:

- Cloud server aggregate edge server's model across all clusters **asynchronously** as:

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

# Contents

# Hardware platform




- When resource-constrained smart meter meet resource-intensive neural network training.
  - **Memory**: 192KB SRAM and 1MB FLASH for numerous parameters. (16GB RAM and 1TB FLASH on PC)
  - **Computation**: Microcontroller with 168MHz frequency for complex computation. (CPU with 4.9GHz frequency)
  - **Communication**: RS485 with 115.2Kpbs rate for frequency transmission. (Wi-Fi with 300Mbps rate)

**Edge intelligence on smart meters is a challenging task!**

# Case studies

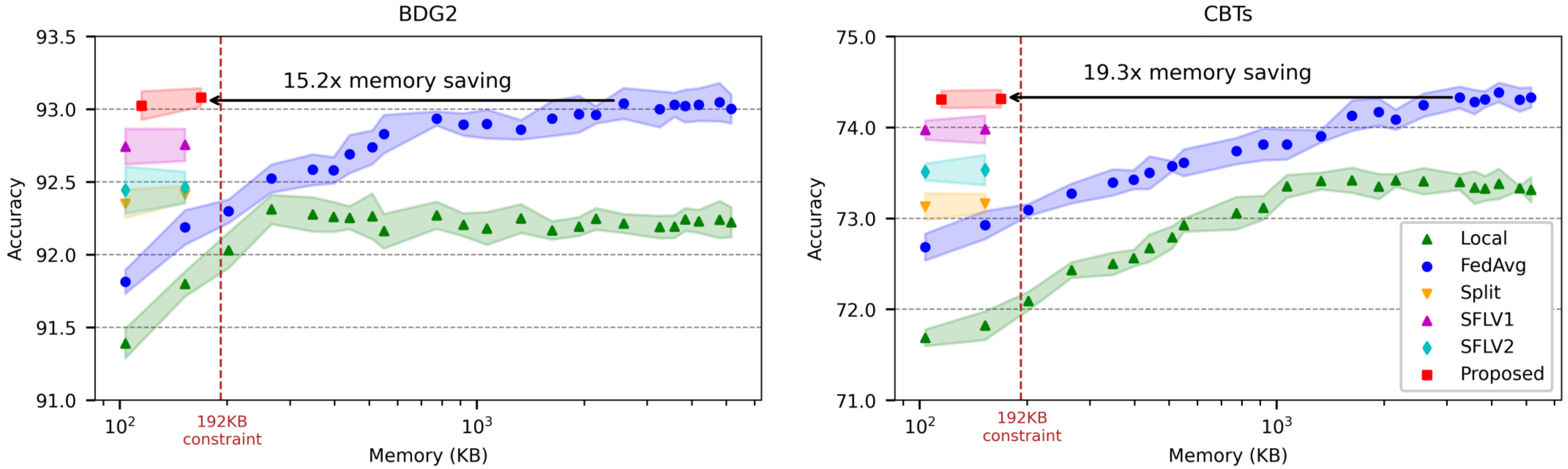
- **Datasets:**
- Building electricity load:
  - BDG2
  - hour-level resolution
  - 206-01-01 to 2017-12-31
- Household electricity load :
  - CBTs
  - hour-level resolution(aggregated by 30min)
  - 2009-07-01 to 2010-12-31

- **Benchmarks:**
- Centralized learning: <u>Cen</u>
- Local learning: <u>Local</u>
- Federated learning: <u>FedAvg</u>
- Split learning: <u>Split</u>
- Federated split learning:
  - <u>SFLV1</u>
  - <u>SFLV2</u>
  - <u>Proposed</u>

- **Metrics:**
- Accuracy metrics: RMSE, MAPE, MAE
- Efficiency metrics: Memory, Training time, Communication overhead

# Results

☐ Comparison of **accuracy versus memory usage** on smart meters with different model sizes




- Compared with the benchmark methods, the proposed method significantly improves the accuracy within the 192KB memory constraint.
- Compared with the benchmark methods, the proposed method saves more than 15x of memory with the same accuracy.

# Results

- Performance of different methods in terms of **accuracy, memory, training time, and communication overhead** per round.

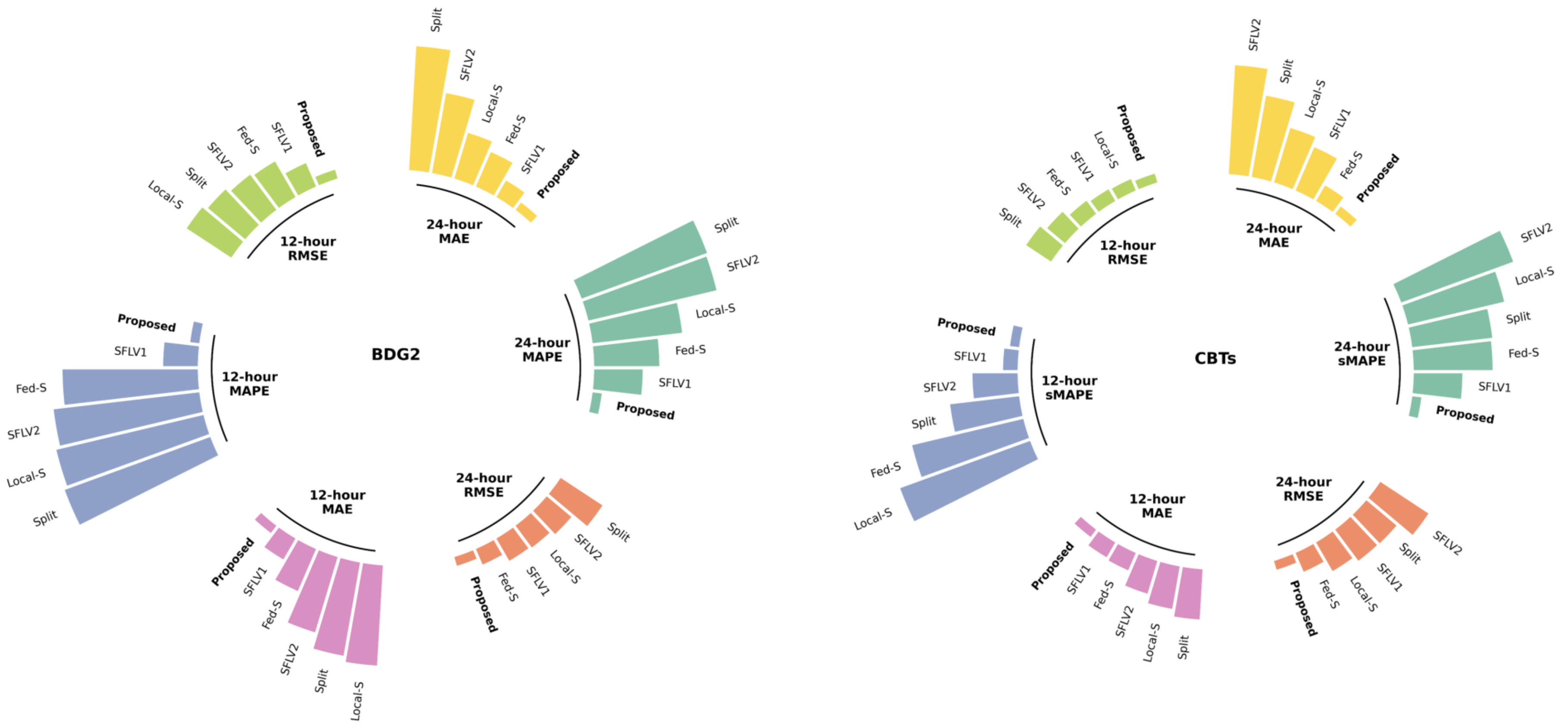
| Method | BDG2 | | | CBTs | | | Memory (KB) | Training Time (s) | Communication (KB) |
|---|---|---|---|---|---|---|---|---|---|
| | RMSE | MAPE | MAE | RMSE | MAPE | MAE | | | |
| Cen | 22.82 | 8.41 | 8.20 | 0.4780 | 26.98 | 0.2727 | 2578.42 (1.0×) | 586.42 (2.05×) | - |
| Local-S | 23.56 | 8.13 | 8.12 | 0.4698 | 28.19 | 0.2726 | 152.53 (16.9×) | 41.19 (29.14×) | - |
| FedAvg-S | 22.79 | 7.67 | 7.98 | 0.4681 | 27.04 | 0.2699 | 152.53 (16.9×) | 46.49 (28.80×) | 5.50 (27.18×) |
| Local-M | 22.84 | 7.75 | 8.03 | 0.4645 | 26.67 | 0.2682 | 2578.42 (1.0×) | 1187.15 (1.01×) | - |
| FedAvg-M | <u>22.25</u> | **6.96** | <u>7.48</u> | **0.4614** | <u>25.81</u> | <u>0.2637</u> | 2578.42 (1.0×) | 1200.44 (1.0×) | 149.50 (1.0×) |
| Split | 23.27 | 7.68 | 7.96 | 0.4679 | 26.83 | 0.2687 | 103.75 (24.8×) | 96.00 (12.50×) | 91.25 (1.63×) |
| SFLV1 | 22.34 | 7.25 | 7.68 | 0.4647 | 26.03 | 0.2664 | 103.75 (24.8×) | 96.49 (12.44×) | 96.75 (1.54×) |
| SFLV2 | 22.76 | 7.53 | 7.89 | 0.4674 | 26.49 | 0.2683 | 103.75 (24.8×) | 96.49 (12.44×) | 96.75 (1.54×) |
| Proposed | **22.17** | <u>6.98</u> | **7.44** | <u>0.4630</u> | **25.74** | **0.2636** | 115.07 (22.4×) | 62.41 (19.23×) | 74.43 (2.01×) |

[1] The best- and the second-best-performed methods are bolded and underlined, respectively.
[2] The -S and -M indicate the model has single and multiple hidden layers, respectively.

- The proposed method reduces 95.6% memory footprint, 94.8% training time, and 50% communication overhead.

# Results

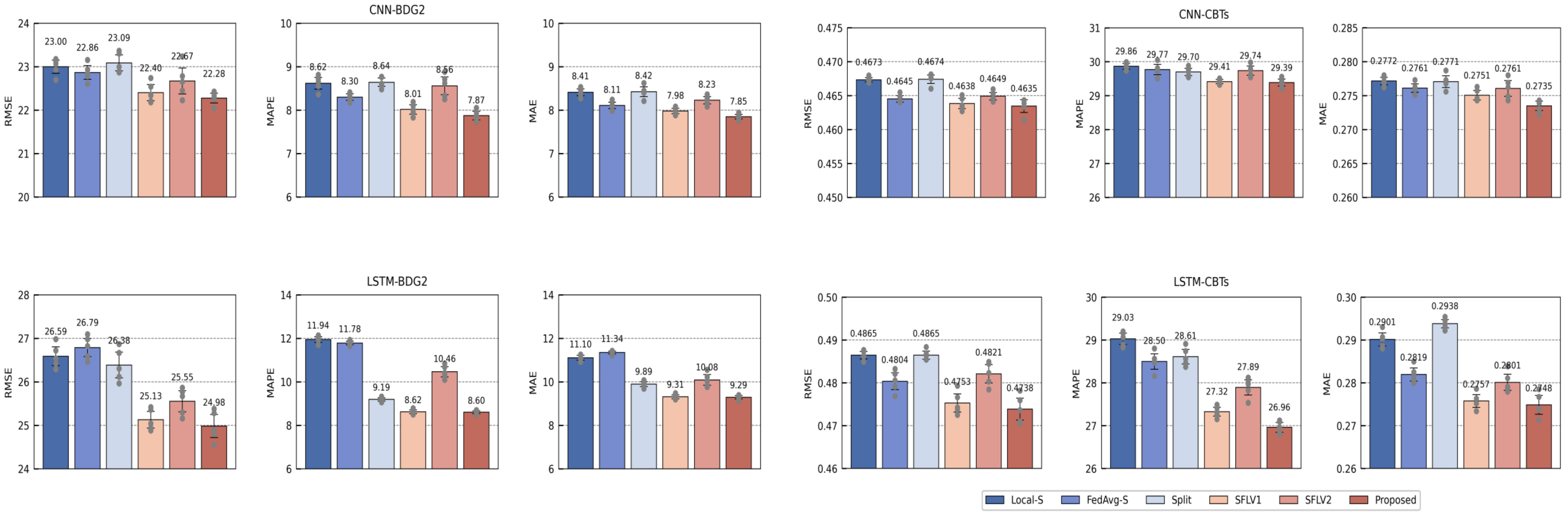
- Performance evaluation of the proposed method on **different forecasting ranges**




- Compared to the best-performing benchmarks in each task, our proposed model has at least 1.33%, 2.19%, and 3.27% improvement in terms of RMSE, MAPE, and MAE, respectively.

# Results

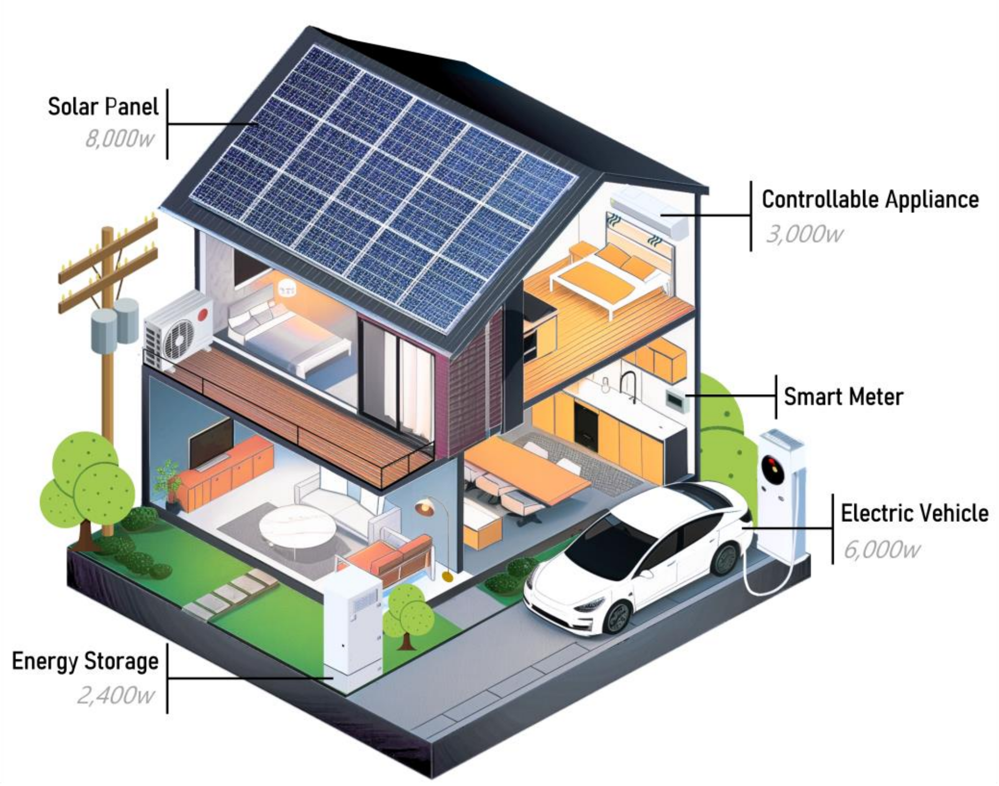
- Performance evaluation of the proposed method with **different neural network backbones**

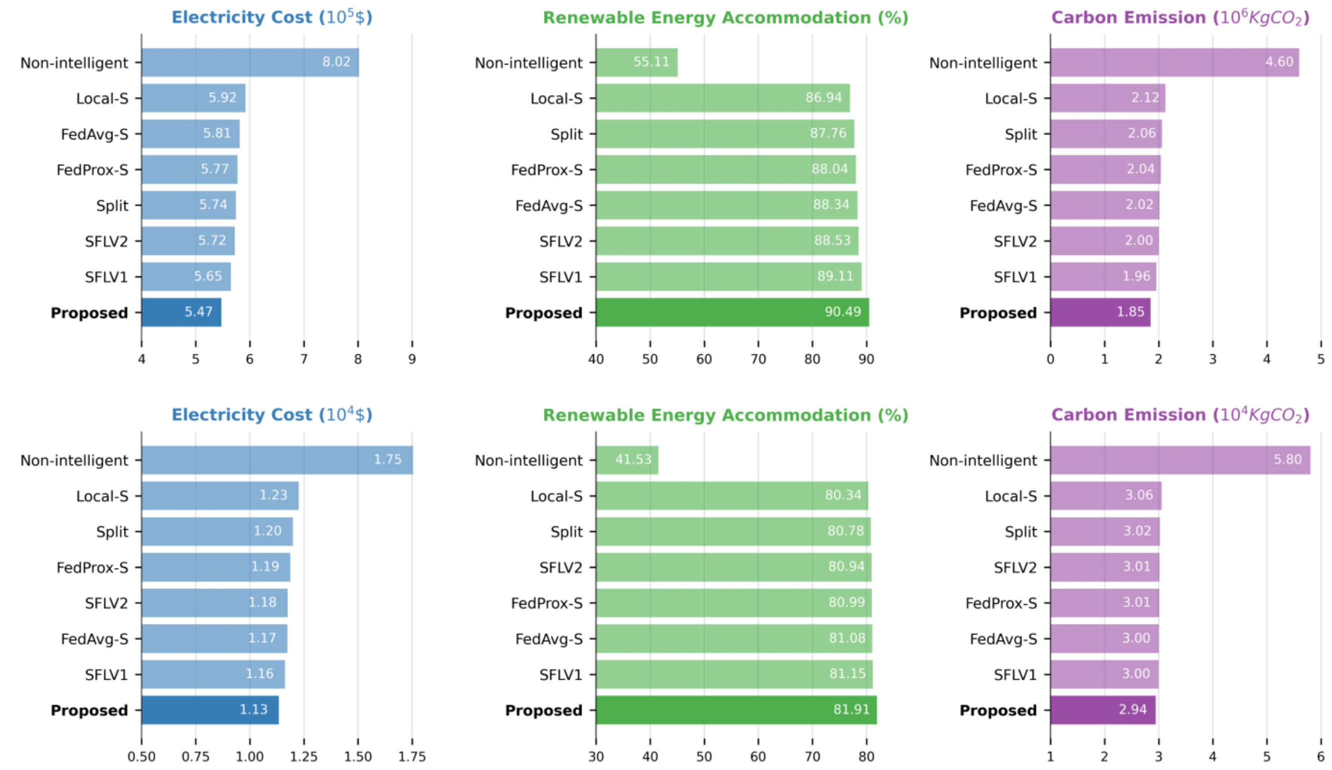



- The proposed method surpasses all other on-device feasible methods with different neural networks as the backbone

# Results

- Comparison of **electric cost, renewable energy accommodation ratio, and carbon emission** for non-intelligent strategy and various edge intelligent methods.

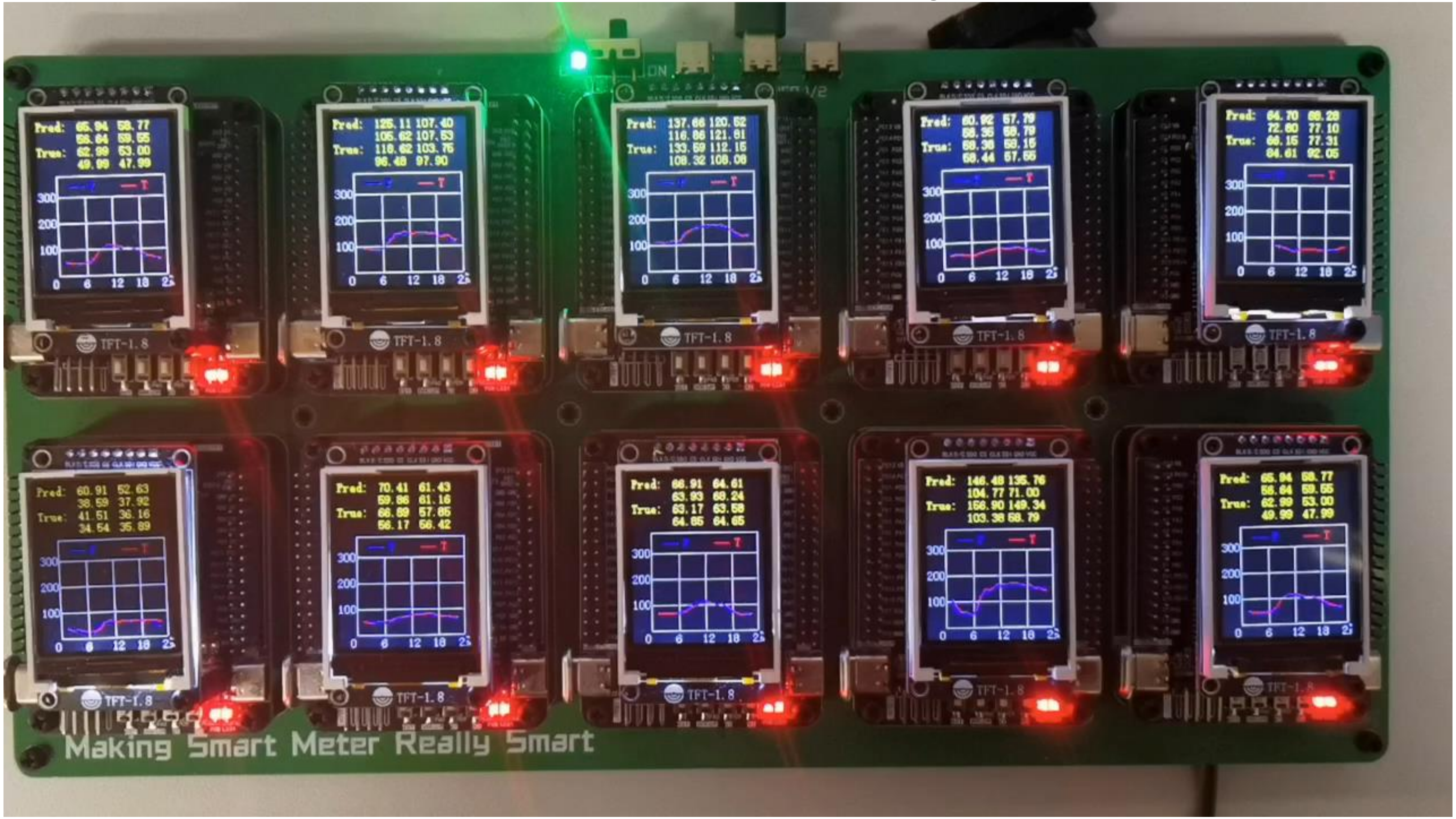



- By adopting our approach, it is expected to save \$1,176.11 in electricity costs per building annually and \$18.93 in electricity costs per household annually.

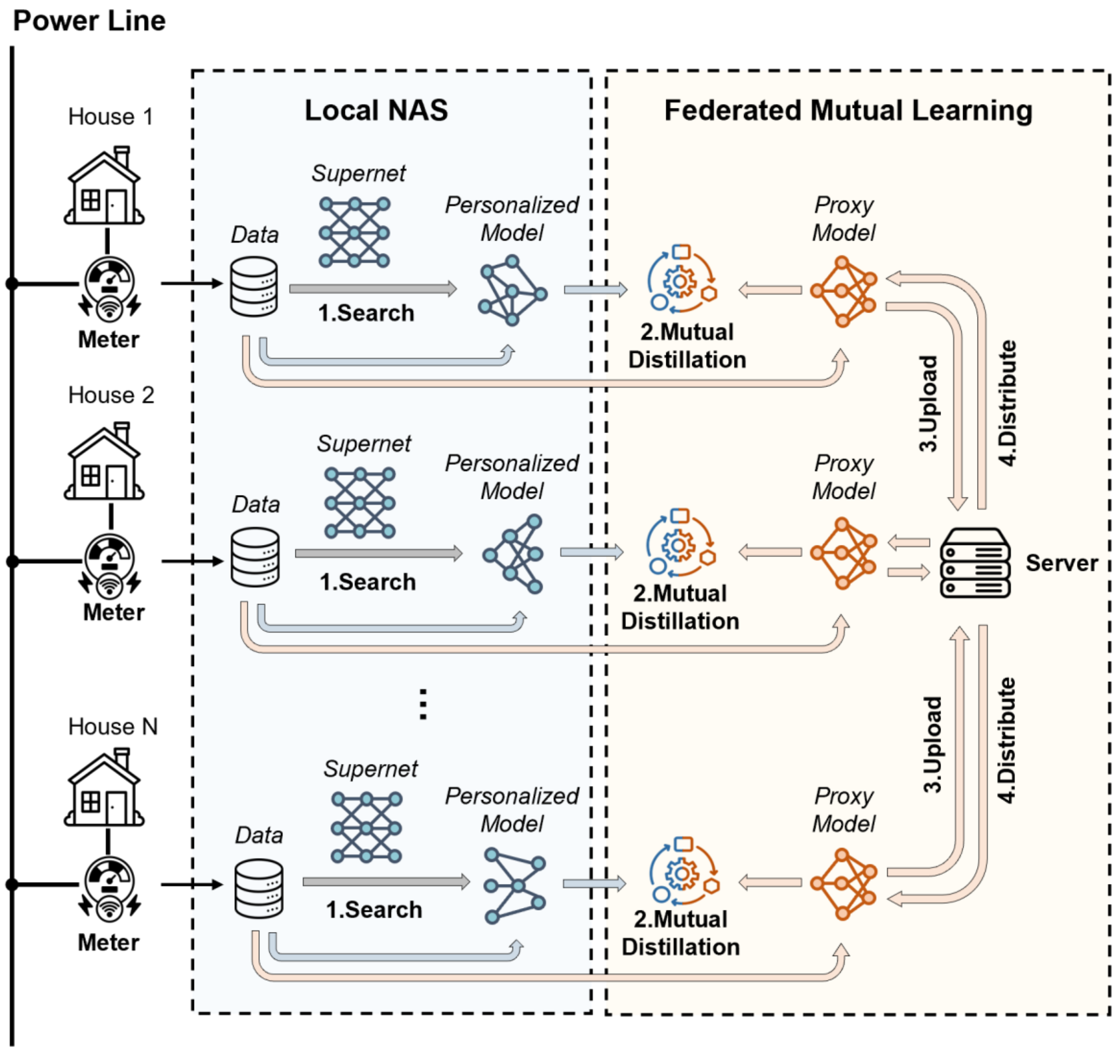
# Results

## 4 hour-ahead Load Forecasting



# Contents

# Monitoring: Framework




**Overview of the lightweight NILM framework**

- The proposed NILM method aims to develop high-performance models on end devices for accurately estimating appliance-level power consumption:

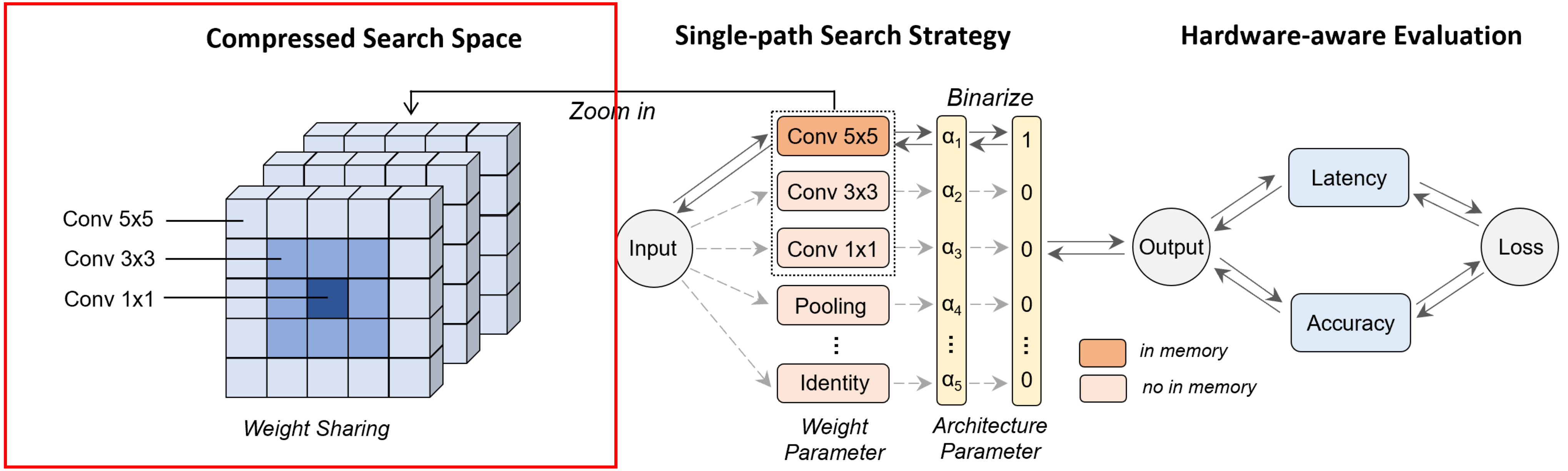
① **Search**: personalized model is automatically searched from a supernet using memory-efficient NAS. The designed architecture varies depending on the appliances and households involved.

② **Mutual Distillation**: unified proxy model with same network architecture transfers global knowledge to and obtains local knowledge from the specialized personalized model via mutual distillation

③ **Upload**: the weight parameters of the proxy models are uploaded to the server for average aggregation

④ **Distribute**: The aggregated weight parameters are distributed to update the proxy model

# Monitoring: Memory-efficient NAS



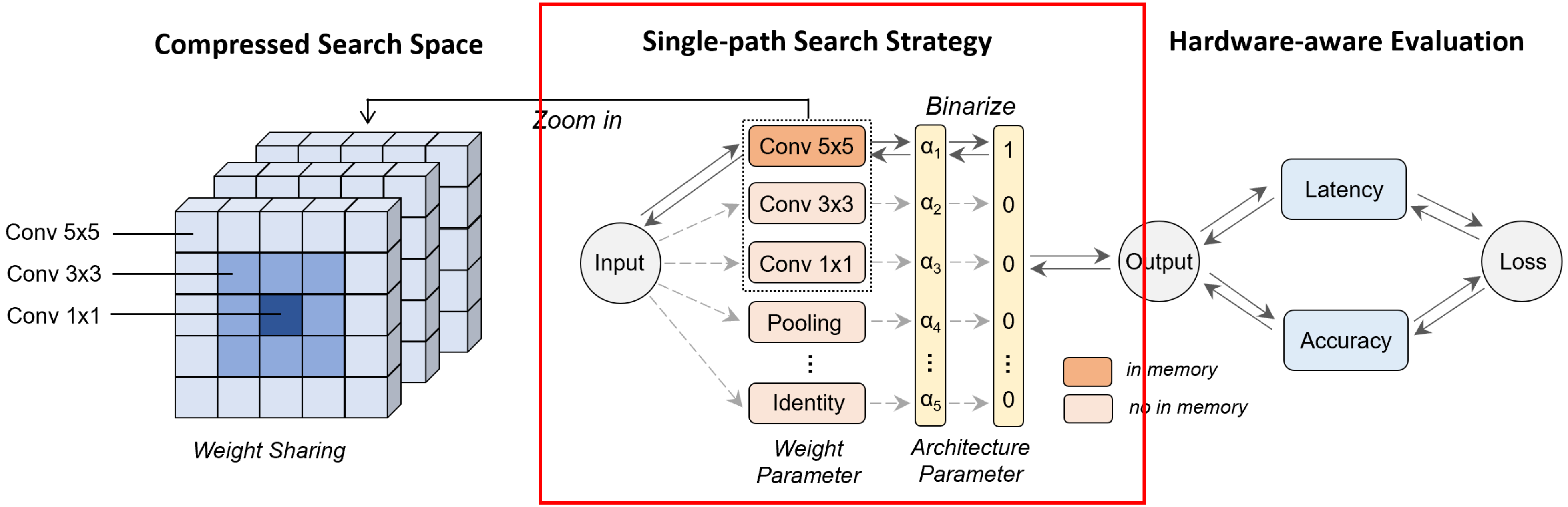

**1. Compressed Search Space**:

- To reduce the memory footprint, we compress all candidate convolutional operations per layer into a single net by sharing the kernel weights. Specifically, the weights of the small kernels can be viewed as the subnet of the large kernels.

$$w_{5\times5} = w_{1\times1} + w_{3\times3|1\times1} + w_{5\times5|3\times3}$$

where $w_{3\times3|1\times1}$ and $w_{5\times5|3\times3}$ denote the 3×3 and 5×5 convolutional kernels excluding subsets, respectively.

# Monitoring: Memory-efficient NAS



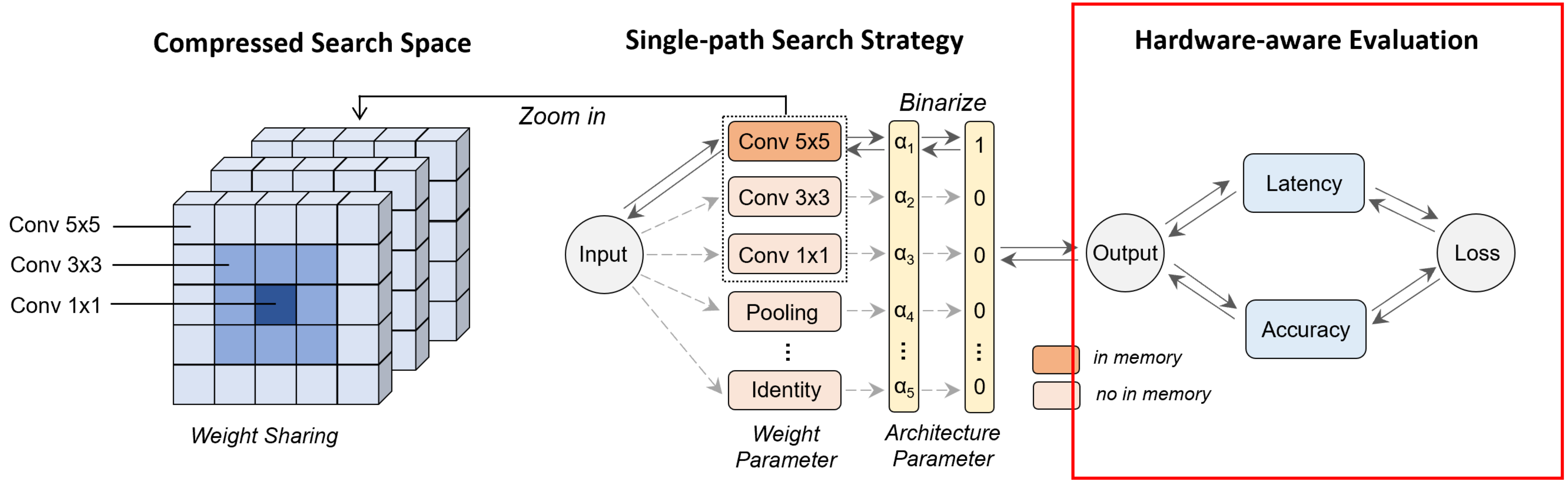

**2. Single-Path Search Strategy**:

- To decrease the memory consumption of architecture search to a level comparable to compact model training, we introduce path binarization, where only one path is selected for each round of updates.

$$g = G(p_1, \ldots, p_M) = \begin{cases} [1, 0, \ldots, 0] & \text{with prob. } p_1 \\ \cdots & \\ [0, 0, \ldots, 1] & \text{with prob. } p_M \end{cases}$$

where $g$ denotes a binary vector and the value of variable $g_i$ indicates whether or not the i-th path is sampled.

# Monitoring: Memory-efficient NAS



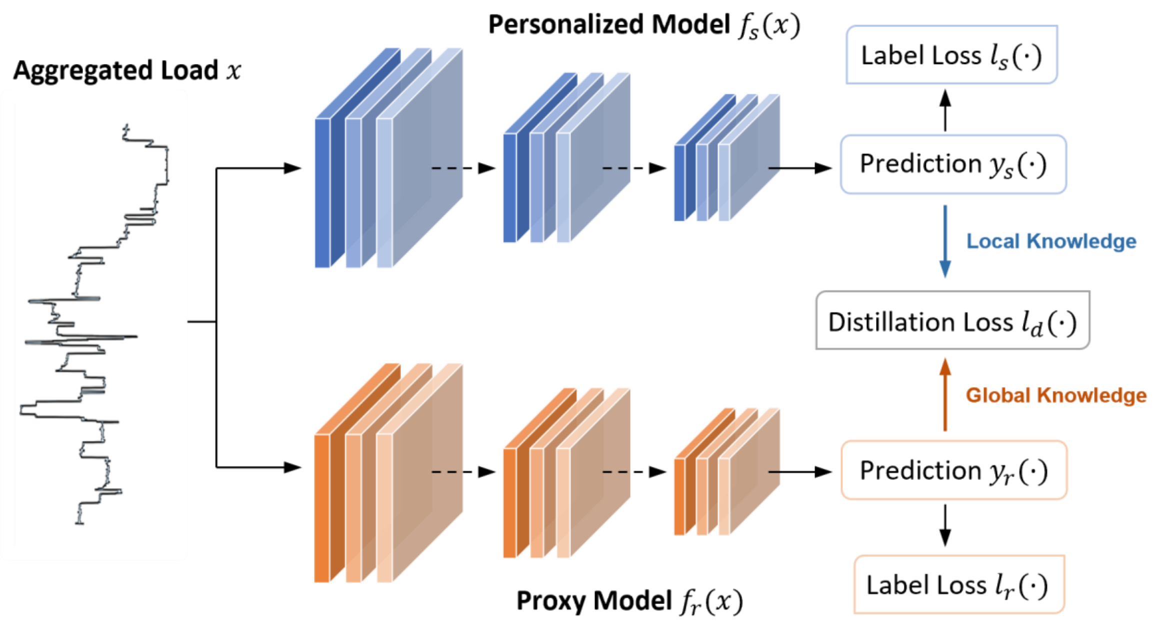

**3. Hardware-Aware Evaluation**:

- Real-time NILM imposes a time constraint on model inference latency to enable continuous monitoring. Hence, we attempt to achieve a trade-off between accuracy and latency when searching for the architecture of the personalized model.

$$\begin{array}{ll} \min\limits_{\alpha} & \mathcal{L}_{\text{val}}(w, \alpha) + \left(\frac{T(\alpha)}{T_0}\right)^{\lambda} \\ \text{s.t.} & w = \operatorname{argmin}_w \mathcal{L}_{\text{train}}(w, \alpha) \end{array}$$

where $T_0$ denotes the time length of the sliding window used in NILM and $T$ denotes the network latency.

# Monitoring: Adaptive Mutual Learning




*To cope with the heterogeneity of personalized models, we adopt mutual distillation for federated learning by introducing unified proxy models with same architecture.*

- Each end device updates the personalized model and proxy model based on both **label loss** and **mutual distillation loss** between the two models.

$$\mathcal{L}_{\text{train},s} = \ell_s(w_s, x) + \ell_d(w_s, w_r, x)$$
$$\mathcal{L}_{\text{train},r} = \ell_r(w_r, x) + \ell_d(w_s, w_r, x)$$

The beneficial information is bidirectionally transferred through knowledge distillation.

- Personalized model has tailored architecture, deeply drawn local knowledge guide proxy model training.
- Proxy models are collaboratively aggregated among various devices, personalized model can benefit from global knowledge of proxy model.

# Monitoring: Results

- Performance evaluation of different methods **on real-world two datasets**.

TABLE I
COMPARISON OF MODEL PERFORMANCE ON THE REFIT DATASET

| Appliance | Metric | Centralized | Local | NAS | MNAS | Federated | Proposed |
|---|---|---|---|---|---|---|---|
| Fridge | MAE | 32.15 | 30.14 | 28.41 | 28.5 | 29.05 | **27.99** |
| | SAE | 1.131 | 1.125 | 1.042 | 1.057 | 1.083 | **1.034** |
| Washing machine | MAE | 18.25 | 23.7 | 19.55 | 19.05 | 21.94 | **17.34** |
| | SAE | 1.508 | 1.781 | 1.742 | 1.525 | 1.821 | **1.460** |
| Dishwasher | MAE | 38.42 | 39.36 | 36.94 | 36.99 | 36.58 | **36.11** |
| | SAE | 1.176 | 1.223 | 1.167 | 1.173 | 1.14 | **1.133** |
| Microwave | MAE | 9.52 | 10.22 | 9.24 | 9.49 | 9.56 | **8.58** |
| | SAE | 1.496 | 1.797 | 1.498 | 1.521 | 1.485 | **1.401** |
| Kettle | MAE | 20.64 | 23.25 | 21.03 | 20.81 | 20.75 | **18.24** |
| | SAE | 1.813 | 2.012 | 1.796 | 1.789 | 1.856 | **1.552** |

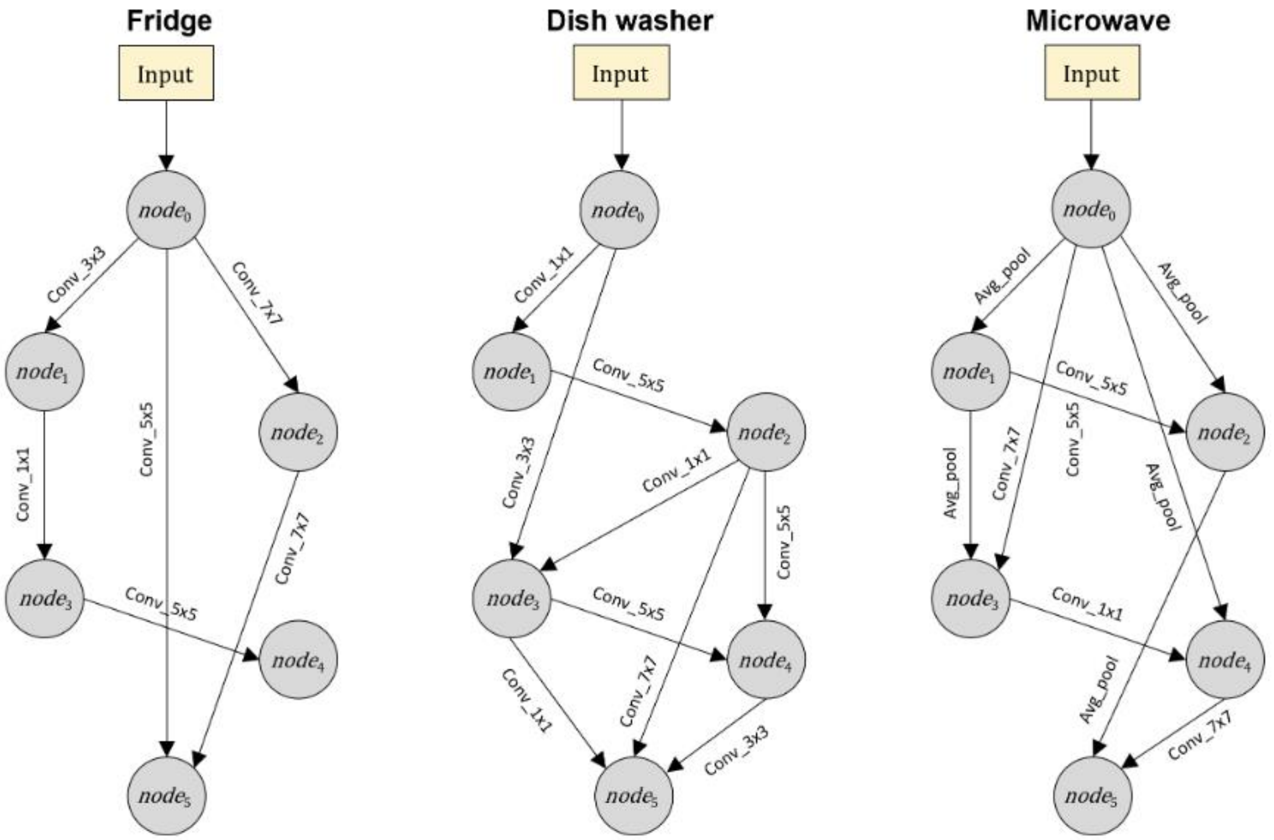
TABLE II
COMPARISON OF MODEL PERFORMANCE ON THE REDD DATASET

| Appliance | Metric | Centralized | Local | NAS | MNAS | Federated | Proposed |
|---|---|---|---|---|---|---|---|
| Fridge | MAE | 32.78 | 34.45 | 32.83 | 32.47 | 32.99 | **31.73** |
| | SAE | 0.475 | 0.496 | 0.478 | 0.471 | 0.476 | **0.462** |
| Dishwasher | MAE | 11.74 | 9.63 | 9.06 | 8.84 | 8.8 | **8.51** |
| | SAE | 1.265 | 1.121 | 1.107 | 1.097 | 1.052 | **1.008** |
| Microwave | MAE | 18.74 | 20.52 | 19.22 | 18.96 | 18.31 | **17.89** |
| | SAE | 1.126 | 1.22 | 1.158 | 1.125 | 1.098 | **1.081** |

- Our proposed model, uniting the strength of architecture search and federation, shows the best performance.

# Monitoring: Results

- The searched network architecture for different appliances.




- For the **fridge**, the architecture of NAS includes few convolutional layers because cyclic power consumption patterns are easily identified.
- By contrast, the searched network for the **dishwasher** integrates more convolutional layers to capture continuously varying power consumption characteristics.
- For the **microwave**, the personalized model contains several pooling layers to focus on localized information of short duration.

# Monitoring: Results

- Comparison of **training and testing efficiency** for different methods.

NAS:

$$\text{Time} \sim O\left(\sum_{l=1}^{D}\sum_{i=1}^{P} K_{l,i} \cdot M_{l,i} \cdot C_{l-1} \cdot C_l\right)$$

$$\text{Space} \sim O\left(\sum_{l=1}^{D}\sum_{i=1}^{P} K_{l,i} \cdot C_{l-1} \cdot C_l + \sum_{l=1}^{D}\sum_{i=1}^{P} M_{l,i} \cdot C_l\right)$$

Proposed:

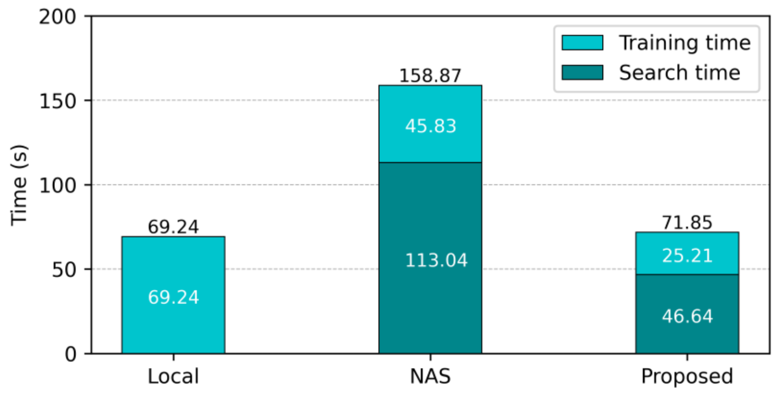
$$\text{Time} \sim O\left(\sum_{l=1}^{D} K_l^* \cdot M_l \cdot C_{l-1} \cdot C_l\right)$$

$$\text{Space} \sim O\left(\sum_{l=1}^{D} K_l^* \cdot C_{l-1} \cdot C_l + \sum_{l=1}^{D} M_l \cdot C_l\right)$$

computational complexity is reduced by *P* times

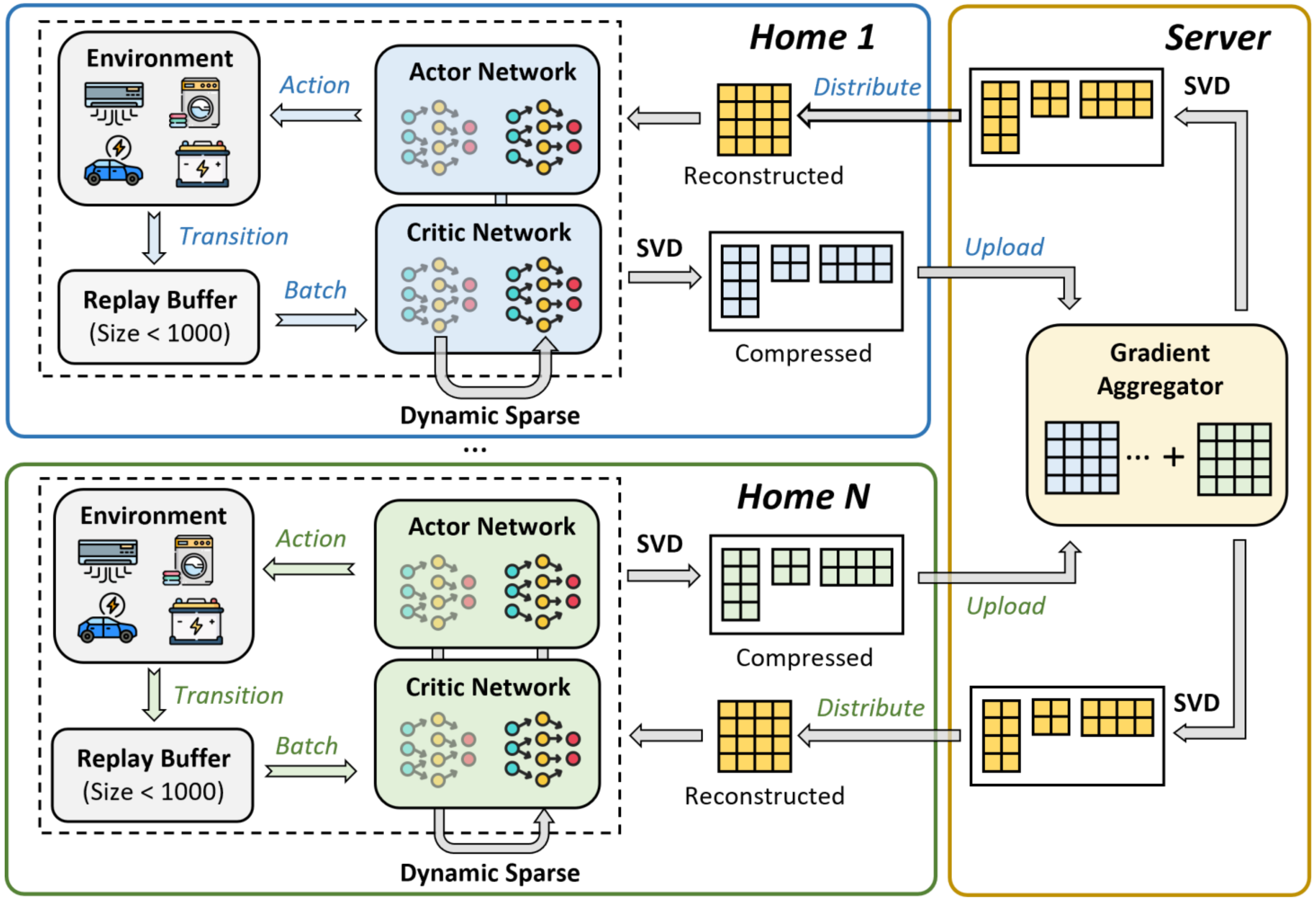
TABLE III
COMPARISON OF TESTING EFFICIENCY FOR DIFFERENT METHODS

| Method | Model size (Kb) | Inference time (s) |
|---|---|---|
| Local | 232.448 | 6.55E-03 |
| NAS | 87.834 | 4.18E-03 |
| Proposed | 56.089 | 2.53E-03 |




The results show that the size of the customized architecture, i.e., the memory space required, is smaller than that of the fixed model. Moreover, the proposed hardware-aware method can decrease disaggregating time by more than 2.5 times.

# Decision-making: Framework



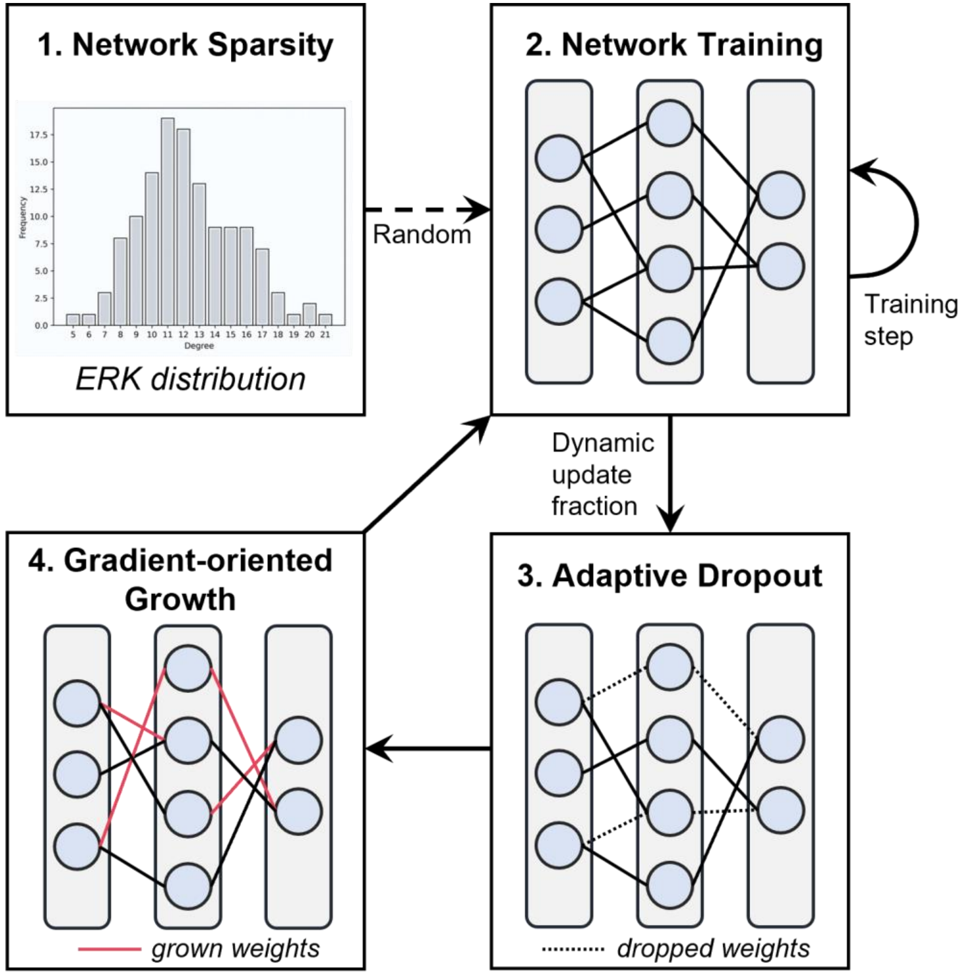

**Overview of the edge HEM framework**

- To enable efficient RL for edge home energy management, our framework incorporates two critical techniques:

① **Dynamic Sparse Learning**: the edge device trains sparse neural networks from scratch and dynamically adapts its topology to the changing data distribution during training

② **Compressed Federate Learning**: the edge devices utilize randomized SVD for gradient approximation to reduce the communication overhead in federated learning

# Decision-making: Dynamic sparse learning




**1. Network Initialization**:

The initial network connection follows the **Erdös–Rényi graph**. The probability of the weight connection between the neurons can be expressed as:

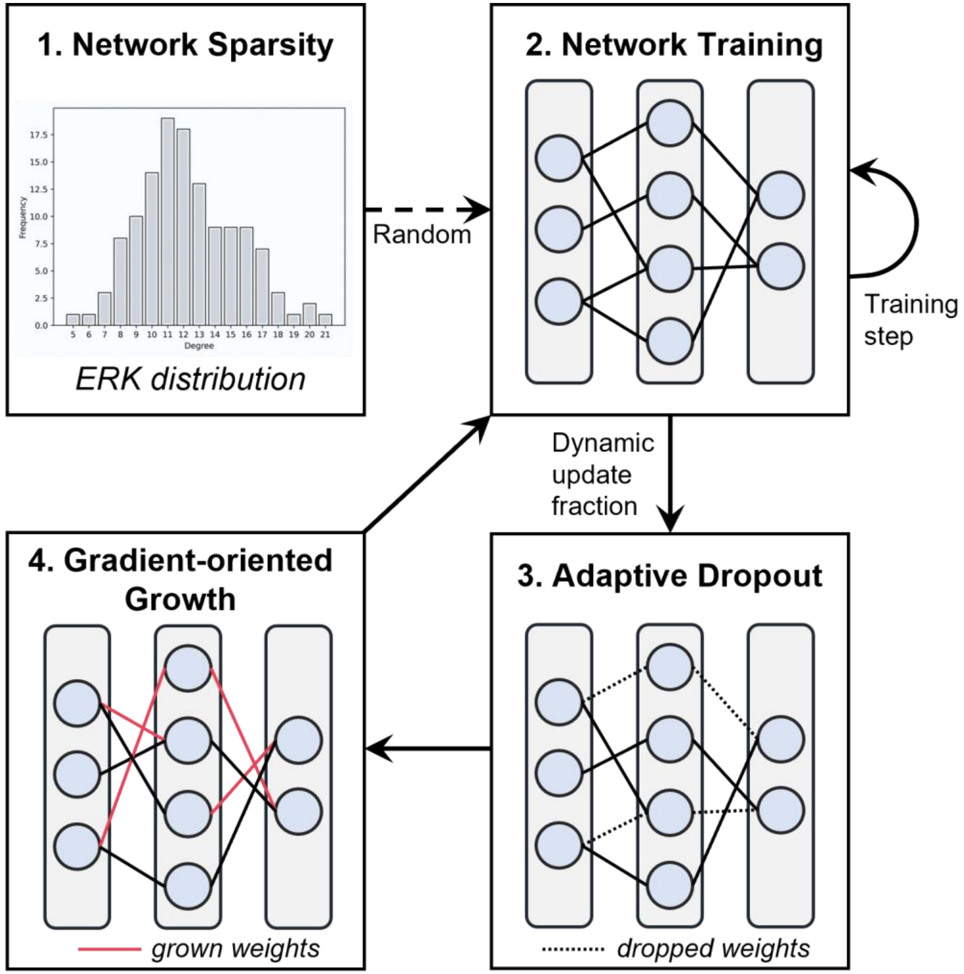
$$p(w_{i,j}^k) = \kappa \cdot \frac{n^k + n^{k+1}}{n^k \cdot n^{k+1}}$$

**2. Network Training**:

The agent learns through trial and error cycles, collecting the data online while interacting with the environment.

The agent perform dynamic adaptation of the sparse topology after each training epoch as follows.

# Decision-making: Dynamic sparse learning




**3. Adaptive Dropout**:

The network **dynamically drop** a fraction of the weight *closest to zero* at each training epoch, which are considered as the least important:

$$\zeta_t = \frac{\zeta_0}{2}\left(1 + \cos\left(\frac{\pi t}{T_{\text{end}}}\right)\right)$$

The fraction of drops decreases gradually with training of reinforcement learning up to convergence

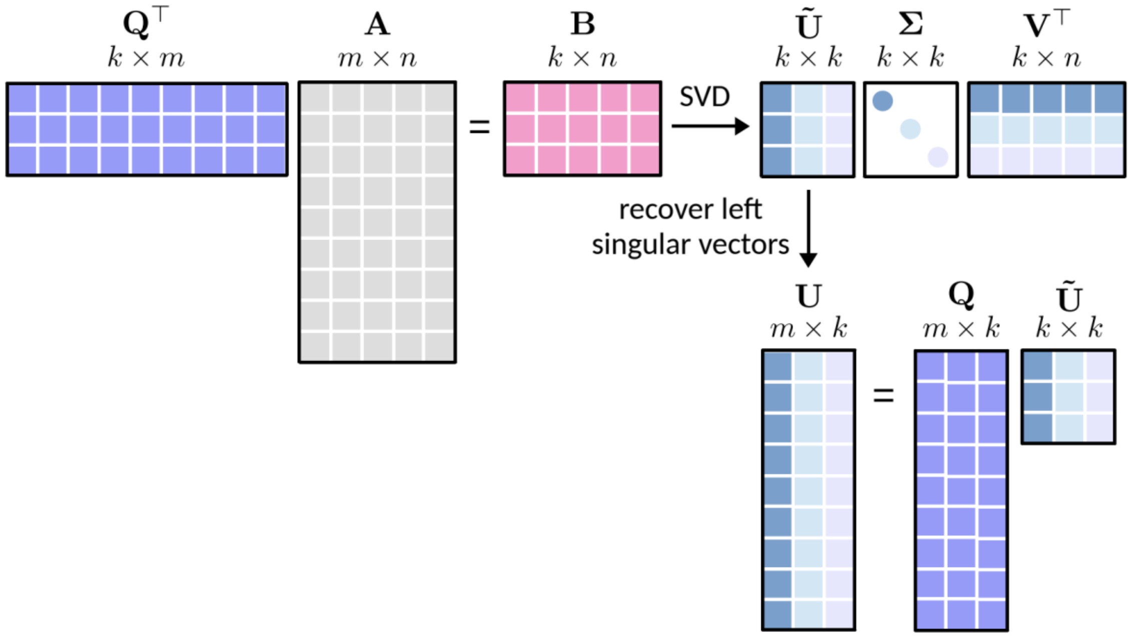
**4. Gradient-oriented Growth**:

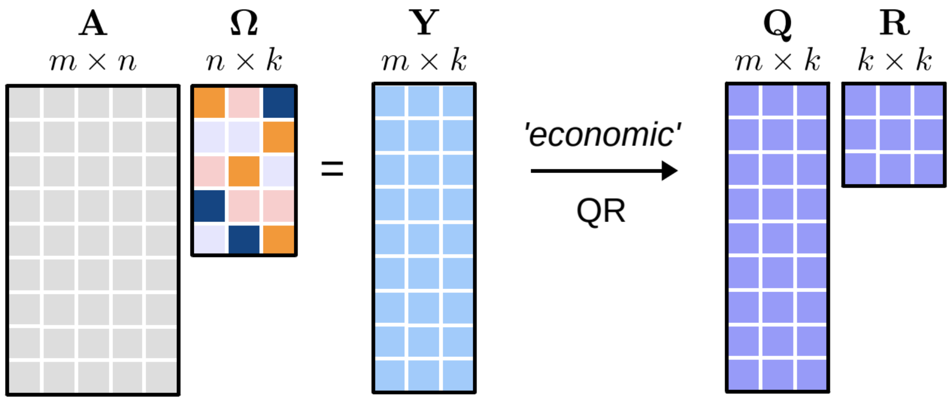
Then**, new weights are added** in the same amount as the ones previously removed based on the *gradient values*.

$$\frac{\partial \mathcal{L}}{\partial w_{i,j}^k} = x_i^k \cdot \frac{\partial \mathcal{L}}{\partial x_j^{k+1}}$$

Basic idea is that the more sensitive the loss is to changes in a weight during training, the more important that weight is.

# Decision-making: Compressed federated learning




- **Sparse characteristic** of our networks enables the gradient, which retains only a small number of singular values, to contain almost all of the information entropy of the matrix.

$$W = \sum_{i=1}^{\min(m,n)} \sigma_i U_i V_i \ .$$

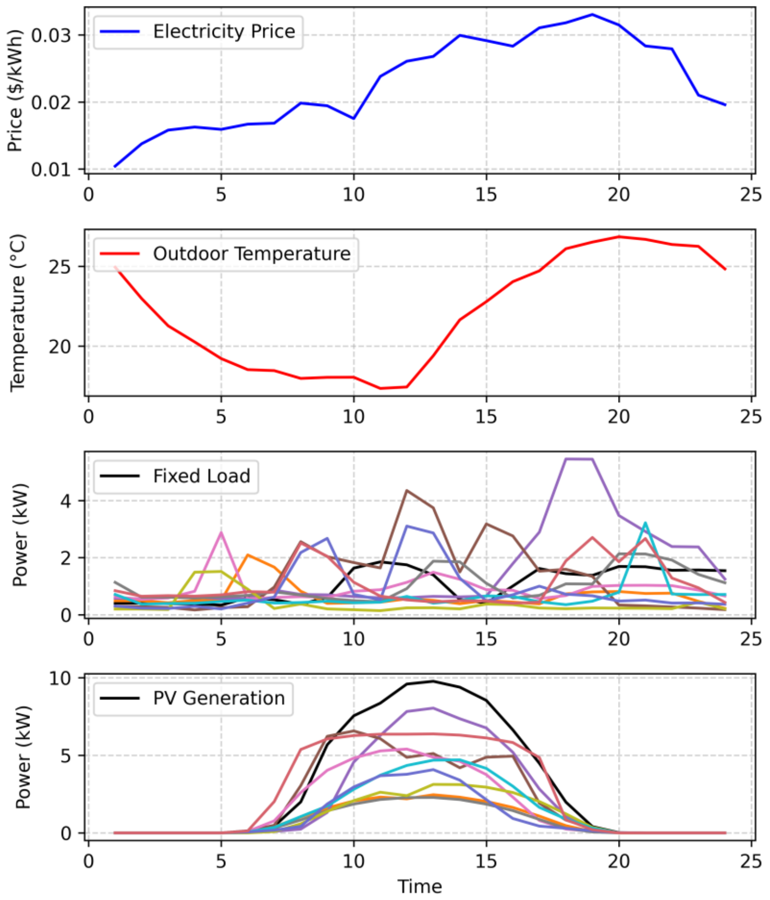
Assuming that only *k* singulars are retained, the optimal approximation of the parameter matrix W is

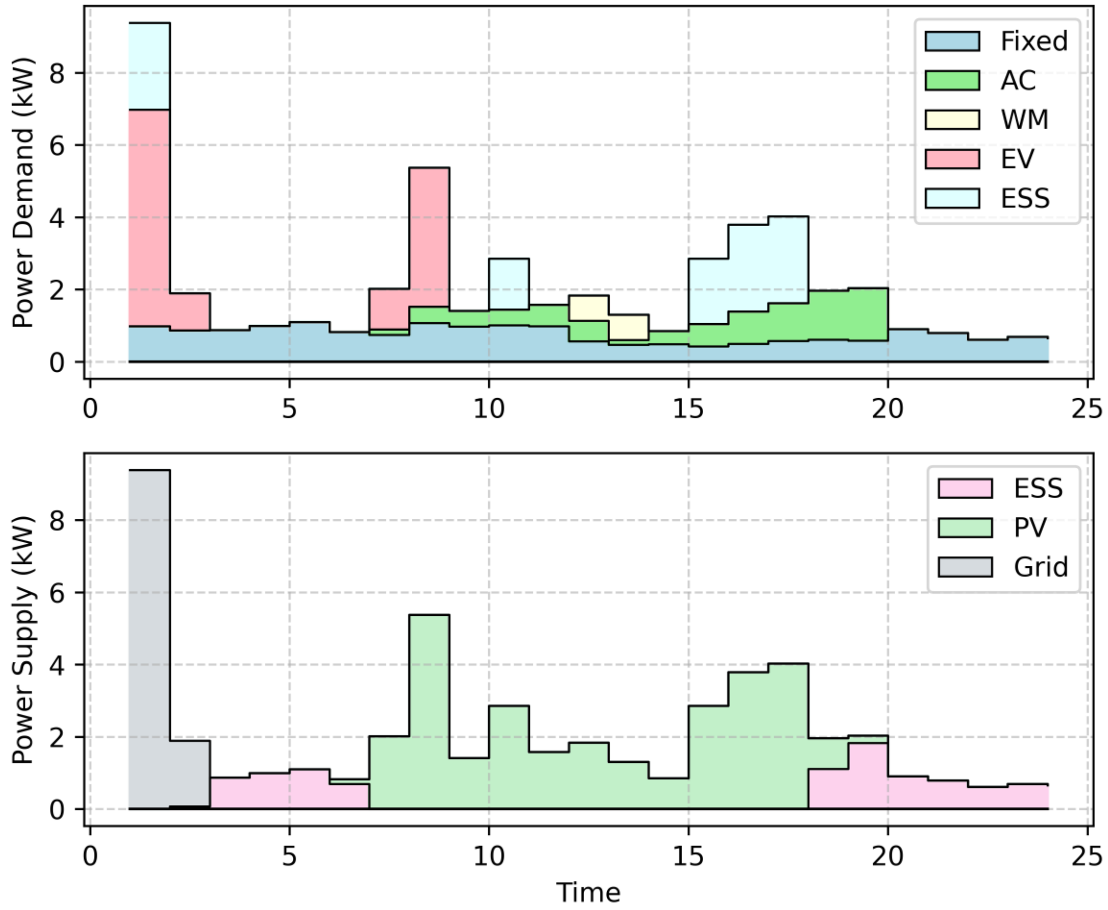
$$\widetilde{W} = \sum_{i=1}^{k} \sigma_i U_i V_i \ .$$

Projecting matrices to lower dimensional linear subspaces greatly reduces costs of federated learning communication.

# Decision-making: Results

- Power **demand and supply** of all loads with proposed scheduling method for 10 homes.




- The proposed method can effectively manage multiple flexible resources to minimize electricity costs.

# Decision-making: Results

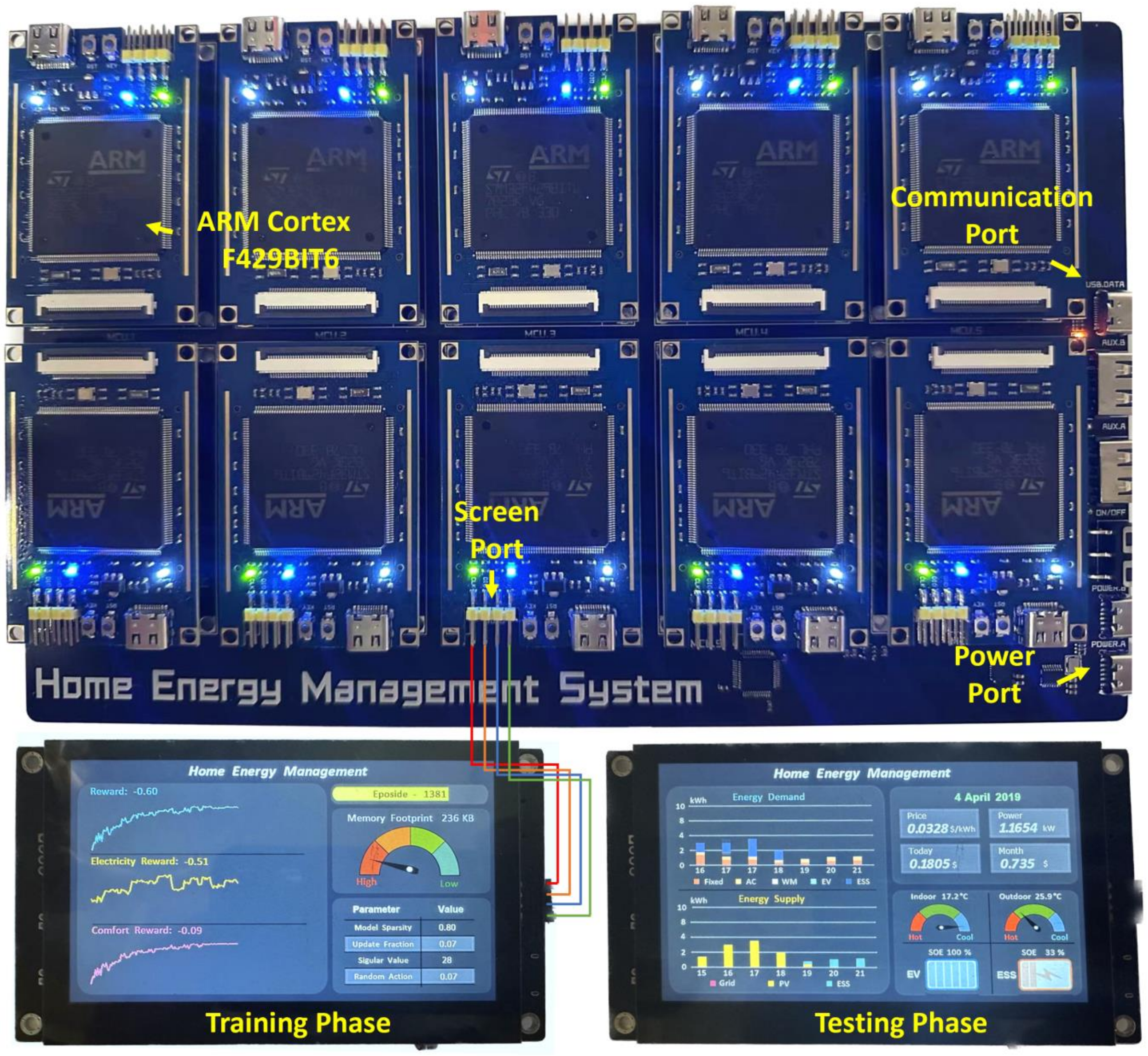
- Performance of different methods in terms of **accuracy, memory, and communication overhead** per round.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODELS IN TERMS OF COST, MEMORY, AND COMMUNICATION

| Model | Cost | | | | | | | | | | | Mem. (MB) | Comm. (MB) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Home1 | Home2 | Home3 | Home4 | Home5 | Home6 | Home7 | Home8 | Home9 | Home10 | Average | | |
| Local | 3.55 | 6.88 | 16.85 | 14.65 | 12.36 | 13.26 | 8.25 | 11.09 | 11.56 | 7.38 | **10.58** | 1.10 | - |
| Sparse-0.3 | 4.09 | 6.50 | 12.86 | 13.21 | 13.67 | 11.17 | 6.33 | 9.42 | 7.53 | 10.16 | **9.49** | 0.78 | - |
| Sparse-0.5 | 4.79 | 6.64 | 18.08 | 12.54 | 11.85 | 12.70 | 5.18 | 9.35 | 7.34 | 10.93 | **9.94** | 0.57 | - |
| Sparse-0.8 | 8.08 | 7.82 | 21.18 | 16.97 | 12.37 | 14.65 | 7.04 | 9.06 | 6.40 | 10.11 | **11.37** | 0.24 | - |
| Federated | 2.09 | 4.68 | 8.42 | 14.8 | 6.81 | 13.53 | 4.67 | 6.3 | 5.48 | 6.27 | **7.30** | 1.10 | 13.77 |
| Proposed-0.3 | 2.77 | 4.68 | 9.81 | 13.32 | 6.95 | 10.72 | 4.52 | 5.35 | 6.85 | 6.72 | **7.17** | 0.78 | 10.68 |
| Proposed-0.5 | 2.67 | 6.02 | 9.99 | 11.88 | 8.25 | 10.83 | 5.04 | 6.06 | 4.87 | 6.89 | **7.25** | 0.57 | 8.55 |
| Proposed-0.8 | 2.72 | 5.98 | 8.27 | 12.31 | 9.26 | 10.38 | 5.14 | 5.93 | 6.83 | 6.3 | **7.31** | 0.24 | 4.37 |
| Optimal | 1.96 | 4.48 | 8.11 | 11.52 | 6.72 | 9.99 | 4.42 | 5.21 | 4.59 | 6.14 | **6.32** | 355.72 | - |

- The proposed method reduces 78.2% memory footprint and 68.3% communication overhead without cost sacrifice

# Decision-making: Results



# Contents

# Discussion

Our research is the **first attempt to achieve complex model training on smart meters**, which can achieve a wide impact and serve broad interests from three perspectives:

- Firstly, our research provides a feasible and efficient approach to exploiting the existing ubiquitous smart meters without the need for additional investment**.**
- Secondly, our research provides new directions for broader edge intelligence applications in the smart grid. This will help consumers better exploit flexible resources to save costs and accommodate more distributed renewable energy, and also enable distribution system operators to better observe the system's status and manage the system to reduce operation costs and improve the reliability of the energy supply
- Thirdly, our research enables the utilization of distributed data in a privacy-preserving manner, which will increase consumers' willingness for smart meter adoption, thus promoting smart meter penetration and contributing to the digitalization and decarbonization of smart grids.

# Thanks for your attention