FED-GAME: PERSONALIZED FEDERATED LEARNING WITH GRAPH ATTENTION MIXTURE-OF-EXPERTS FOR TIME-SERIES FORECASTING

Yi Li[1], *Han Liu*[2], *Mingfeng Fan*[3], *Guo Chen*[4], *Chaojie Li*[5*], *Biplab Sikdar*[2]

[1] School of Automation, Central South University
[2]Department of Electrical and Computer Engineering, National University of Singapore
[3]Department of Mechanical Engineering, National University of Singapore
[4] School of Electrical Engineering and Telecommunications, University of New South Wales
[5] Department of Electrical Engineering, City University of Hong Kong

ABSTRACT

Federated learning (FL) on graphs shows promise for distributed time-series forecasting. Yet, existing methods rely on static topologies and struggle with client heterogeneity. We propose Fed-GAME, a framework that models personalized aggregation as message passing over a learnable dynamic implicit graph. The core is a decoupled parameter difference-based update protocol, where clients transmit parameter differences between their fine-tuned private model and a shared global model. On the server, these differences are decomposed into two streams: (1) averaged difference used to updating the global model for consensus (2) the selective difference fed into a novel Graph Attention Mixture-of-Experts (GAME) aggregator for fine-grained personalization. In this aggregator, shared experts provide scoring signals while personalized gates adaptively weight selective updates to support personalized aggregation. Experiments on two real-world electric vehicle charging datasets demonstrate that Fed-GAME outperforms state-of-the-art personalized FL baselines.

***Index Terms*—** Personal federated learning, timeseries forecasting, implicit graph attention, mixture-of-experts

1. INTRODUCTION

Federated Learning (FL) enables collaborative training over decentralized data [1], but its performance is often degraded by statistical heterogeneity. This problem is particularly severe in distributed time-series forecasting tasks such as EV charging demand prediction, where each client exhibits distinct temporal patterns and seasonality. Personalized FL (PFL) addresses this challenge by tailoring models to local characteristics through techniques such as proximal regularization, clustering, or multi-task learning [2, 3, 4, 5, 6]. However, many of these approaches emphasize local adaptation (e.g., extended local training or personalized regularization) to improve client performance, while overlooking a core aspect of FL: the aggregation of knowledge across clients. Since aggregation directly impacts the quality of personalization under heterogeneous data, recent works propose Federated Graph Learning (FGL), enabling clients to collaborate more selectively with similar peers rather than relying on uniform aggregation. [7, 8, 9, 10]. For instance, [7] designed bilevel optimization schemes to train both local models and the GNN model with two types of objective functions. [8] leveraged geographical proximity as client graph, to guide the model aggregation process. However, existing FGL and other advanced PFL approaches face two fundamental challenges.

(1) Imprecise Client Graphs: Existing FGL methods often rely on predefined, static graphs (e.g., based on geography) [10, 11], which may not capture the true task-specific relationships between clients. Some studies attempt to infer dynamic graphs from full model updates [12], but this requires clients to transmit their entire models, and the resulting graphs are typically based on coarse-grained, global similarity measures rather than fine-grained, personalized ones.

(2) Suboptimal Aggregation of Mixed Updates: In conventional PFL frameworks, each client only exchanges a single update vector, which conflates global consensus with client-specific personalization [13]. For example, FedAvg [1] simply averages the updates, which weakens general knowledge and obscures personalized signals. The resulting global model may fail to serve different client effectively. Recent parameter decomposition techniques [14] addresses this by using parameter additive decomposition, extracting global parameters for knowledge sharing while retaining local parameters for personalization. However, this decomposition neglects the rich inter-client relationships that are crucial for truly effective personalization.

To track these challenges, we propose personalized Federated Learning with Graph Attention Mixture-of-Experts (Fed-GAME), a novel PFL framework for time-series forecasting. Our contributions are: (1) We separate global consensus from personalization by transmitting only the parameter differences between each client's fine-tuned private model

*Corresponding author: Chaojie Li (chaojili@cityu.edu.hk)

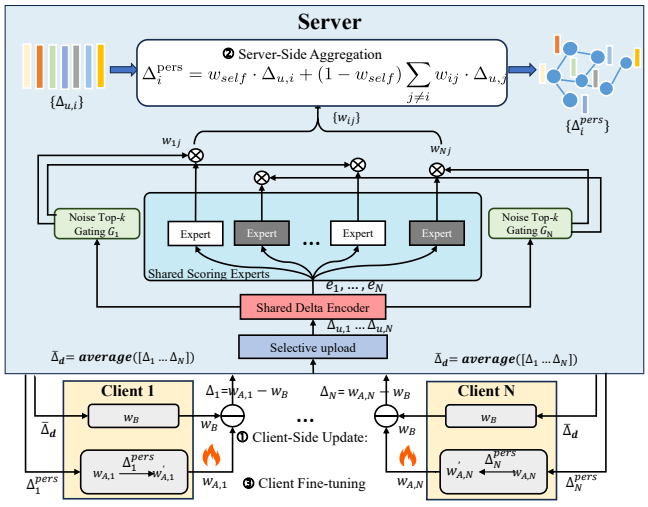

Fig. 1. The Framework of Fed-GAME

(M_A) and the shared global model (M_B). This design enables fine-grained personalized updates and maintain robust global consensus. (2) We propose a server-side GAME aggregator that learns dynamic, client-specific aggregation strategies from the transmitted updates, allowing GAME capture task-specific relationships in a flexible, data-driven manner. (3) We design a similarity-based meta-loss to train the aggregator in server without direct access to client data.

2. THE FED-GAME FRAMEWORK

2.1. Problem Formulation

The goal of this paper is personalized time-series forecasting for N clients in federated setting. Each client $i \in \{1, \cdots, N\}$ possesses a local private dataset $D_i = \{(\mathbf{X}_i, \mathbf{Y}_i)\}$, where $\mathbf{X}_i = \{X_i^{t+1}, \cdots, X_i^{t+h}\}$ represents the historical sequence of length h, and $\mathbf{Y}_i = \{y_i^{t+h+1}, \cdots, y_i^{t+h+p}\}$ is the p-step corresponding future sequence to be predicted. The forecast series of p-steps of i-th client is denoted as $\hat{\mathbf{Y}}_i^p = \{\hat{y}_i^{t+h+1}, \cdots, \hat{y}_i^{t+h+p}\}$. Due to data heterogeneity, instead of learning a single global model, Fed-GAME aims to learn a unique set of model parameters $\boldsymbol{w}_i$ for each client. The objective is to minimize the aggregated loss over all clients: $\min_{\boldsymbol{w}} \sum_{i \in N} \lambda_i f_i(\boldsymbol{w}; \mathcal{D}_i)$, where N denotes the number of clients, $f_i(\cdot)$ is the forecasting loss for client i, λ_i reflects the dataset weight, $\boldsymbol{w}$ is the model parameters to be optimized.

2.2. Fed-Game Framework

To realize personalized time-series forecasting, we propose Fed-GAME. Fed-GAME involves the following steps, each labeled by stage number in Fig. 1 for clarity:

(1) Client-Side Update: Each client i maintains its own persistent private model (M_A) with parameters $w_{A,i}^\tau$ and a local copy of the shared global model (M_B) with parameters w_B^τ in round τ. After fine-tuning M_A on its local data in round τ, client i computes the parameter difference relative to the global model: $\Delta_i^\tau = w_{A,i}^\tau - w_B^\tau$. The client then uploads Δ_i^τ to the server.

(2) Server-Side Aggregation: On the server, these difference are used for two purposes: i. Global consensus update: To maintain a shared knowledge base, the server averages client Δ_i^τ and applies them to the global model: $\overline{\Delta}_d^\tau = \frac{1}{N}\sum_i \Delta_i^\tau$, $w_B^{\tau+1} = w_B^\tau + \eta \cdot \overline{\Delta}_d^\tau$. Here, η is the server learning rate controlling the step size of the global update. ii. Personalized aggregation: For communication efficiency, only selective components of Δ_i^τ (denoted as $\Delta_{u,i}^\tau$) are utilized for personalized aggregation. Here, we restrict to select the final MLP layers, as they are widely recognized to capture high-level transferable features, whereas lower layers remain client-specific. Then, each client's selective difference $\Delta_{u,i}^\tau$ is encoded by a shared Encoder (constructed by Linear layer) into a low-dimensional embedding $\mathbf{e}_i^\tau = \textit{Encoder}(\Delta_{u,i}^\tau)$ to stabilize training and reduce dimensionality for efficient attention computation. These embeddings are then fed into a GAME aggregator, where shared experts model inter-client relevance and personalized gates adaptively weight them. The resulting scores w_{ij}^τ serve as aggregation weights for M_A update, yielding the personalized update $\Delta_i^{\text{pers},\tau}$ as:

$$\Delta_i^{\text{pers},\tau} = w_{self} \cdot \Delta_{u,i}^\tau + (1 - w_{self}) \sum_{j \neq i} w_{ij}^\tau \cdot \Delta_{u,j}^\tau, \quad (1)$$

where w_{self} controls the balance between a client's local updates and the aggregated knowledge from its peers. This corresponds to graph-based message passing on the client interaction graph, where edges encode similarity in final-layer updates. The computation of the graph attention weights $\{w_{ij}^\tau\}$ is detailed in Section 2.3.

(3) Client Fine-tuning: Each client applies the personalized $\Delta_i^{pers,\tau}$ and fine-tunes on local data:

$$w_{A,i}^{\tau+1} = \text{FineTune}(w_{A,i}^{\tau'}) = \text{FineTune}(w_{A,i}^\tau + \gamma \cdot \Delta_i^{\text{pers},\tau}), \quad (2)$$

γ denotes the learning rate controlling the step size of the personalized update. After fine-tuning, the next-round $\Delta_i^{\tau+1}$ is computed as $\Delta_i^{\tau+1} = w_{A,i}^{\tau+1} - w_B^{\tau+1}$ for next round upload.

2.3. GAME Aggregator for Aggregation Weights

The core of our server-side personalization is a learnable Mixture-of-Experts (MoE) aggregator. In each round (superscript τ omitted for clarity), we construct a fully connected implicit graph where each uploaded update $\Delta_{u,i}$ is treated as a node. Its low-dimensional embedding $\mathbf{e}_i$ serves as the dynamic node feature, while the raw update $\Delta_{u,i}$ is propagated as the message. Unlike approaches with fixed adjacency matrices, our aggregator learns personalized graph attention coefficients $\{w_{ij}\}_{j \neq i}$, which adaptively weight messages

(raw update $\Delta_{u,i}$) for each client. This personalized attention mechanism is realized through two components: shared scoring experts, which capture inter-client relevance, and personalized gating modules, which adaptively weight expert outputs for each client.

(1) Shared Scoring Experts: We employ M shared scoring experts $\{E_k\}_{k=1}^{M}$ (construct by linear networks). The role of these experts is to evaluate the relevance of neighbor node j's message to a target client i. The scalar score s_{ijk} is calculated as: $s_{ijk} = E_k([\boldsymbol{e}_j, \boldsymbol{e}_i])$. This score represents the assessment of a specific expert regarding the relevance between each pair of clients.

(2) Personalized Noisy Top-k Gating: Each client i is equipped with a personalized gate G_i, which assigns combination weights over experts. The gate generates logits $H_i(e_i) = (e_i \cdot w_{gi}) + \text{SN}(\text{softplus}((e_i \cdot w_{\text{noise}i})))$, where w_{gi} and $w_{\text{noise}i}$ are learnable parameters, SN adds Gaussian noise for exploration, and $\text{softplus}(z) = \log(1 + e^z)$. Applying Top-$k$ selection followed by softmax produces a sparse weight vector: $c_i = G_i(e_i) = \text{Softmax}(\text{Top-}k(H_i(e_i), k))$, which combines expert scores into a single relevance value: $v_{ij} = c_i^\top s_{ij}, s_{ij} = [s_{ij1}, \ldots, s_{ijM}]$. Graph attention coefficients are then obtained via temperature-scaled softmax: $w_{ij} = \exp(v_{ij}/T) / \sum_{l \neq i} \exp(v_{il}/T)$, where T is the temperature. Thus, w_{ij} defines the edge weights of a dynamic, client-centric graph for each round, computed via weighted message passing in Eq. (1); w_{ij} can also be interpreted as personalized weight. details can be found in [15].

Remark 1 *Compared with standard graph attention aggregators (e.g., GAT), Fed-GAME eliminates reliance on static topology and adopts a MoE paradigm, decomposing the single attention function into multiple shared experts and personalized routers, enabling both multi-expert relationship modeling and client-specific aggregation strategies.*

2.4. Training Objectives

The Fed-GAME framework employs a decoupled, two-level optimization strategy. **Client-Side**: Each client i optimizes its private model M_A by minimizing a local objective function, which combines the forecasting loss on its local data with a proximal term from FedProx [2] to regularize local training and mitigate drift from the global model M_B. The local training of client i is formulated as: $\min_{w_A} \left(\mathcal{L}^i_{\text{task}}(w_A) + \frac{\mu}{2} |w_A - w_B|_2^2 \right)$. **Server-Side**: The server trains its learnable aggregation module (Encoder and MoE) using a similarity-based meta-loss L_{meta}, which is defined as a weighted combination of a Mean Squared Error (L2) loss and a cosine dissimilarity loss:

$$\begin{aligned} \mathcal{L}^i_{\text{meta}}(w_{\text{enc}}, w_{\text{moe}}) =& \alpha \, |\Delta_i^{pers} - \Delta_{u,i}|_2^2 \\ &+ \beta \left(1 - \frac{\Delta_i^{pers} \Delta_{u,i}}{|\Delta_i^{pers}||\Delta_{u,i}|} \right) \end{aligned} \tag{3}$$

Table 1. Evaluation Metrics

$$\text{QS} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(1-q)\|y_i^t - \hat{y}_{i,q}^t\|, & y_i^t < \hat{y}_{i,q}^t \\ \frac{1}{n}\sum_{i=1}^{n} q\|y_i^t - \hat{y}_{i,q}^t\|, & y_i^t \geq \hat{y}_{i,q}^t \end{cases}$$

$$\text{ICP} = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1, & \hat{y}_{i,q}^t \leq y_i^t \leq \hat{y}_{i,q'}^t \\ 0, & \text{otherwise} \end{cases}$$

$$\text{MIL} = \frac{1}{n}\sum_{i=1}^{n} \|\hat{y}_{i,q}^t - \hat{y}_{i,q'}^t\|$$

where α and β are balancing hyperparameters, w_{enc} and w_{moe} denote the parameters of the Encoder and MoE. This meta-loss encourages the MoE to generate personalized updates that align with clients' own raw parameter difference, improving personalization while maintaining global consistency.

2.5. Communication Cost Analysis

Communication overhead is a critical bottleneck in FL. Therefore, we analyze the communication cost per round of Fed-GAME and compare it with standard FL baselines like FedAvg and FedProx that exchange full model parameters. Let θ be the total number of parameters in the model. For baselines, both upload and download involve full parameters, and the total communication cost is $C_{baseline} = 2N|\theta|$. For Fed-GAME, let r be the fraction of parameters selected for communication $0 < r < 1$. The total upstream cost is $N \cdot |\theta|$. For the downstream, the server sends a global parameter difference ($\overline{\Delta}_d$) and a personalized parameter difference (Δ_i^{pers}). The total communication cost is $C_{\text{Fed-GAME}} = 2N|\theta| + N \cdot r|\theta| = (2 + r)N|\theta|$. By comparing this with the baseline, we derive the communication cost ratio: Cost Ratio $= \frac{C_{\text{Fed-GAME}}}{C_{\text{Baseline}}} \approx \frac{(2+r)N|\theta|}{2N|\theta|} = 1 + \frac{r}{2}$. The personalized head is tiny ($r < 1\%$), so, the overhead is negligible while enabling stronger server-side MoE aggregation.

3. EXPERIMENTAL RESULTS

We conduct case studies on two real-world datasets, Palo Alto [16] and Shenzhen [17], which capture small and large scale non-IID spatial-temporal demand patterns. The Palo Alto dataset contains 8 charging stations with demand (kWh) recorded at 5-minute intervals, while the Shenzhen dataset includes 247 stations with 30-minute resolution.

We compare Fed-GAME with SOTA, including FedAvg [4], FedProx [2], pFedMe [5], PAG-FedAVG [18] and GCRN-FedAvg [19], where FedProx and pFedMe are PFL, while PAG-FedAVG and GCRN-FedAVG are FGL. We also consider No_FL which trains a forecasting model locally on each client without any communication. For the uncertainty of the charging demand, we use quantile regression for time-series forecasting [20]. Model performance is evaluated using three standard criteria: Quantile Score (QS), Mean Interval Length (MIL), and Interval Coverage Percentage (ICP), summarized in Table 1. For the Palo Alto dataset, we use 3 hours historical data to predict the next 30 minutes (horizon 6) and 60 minutes (horizon 12) charging demand. For Shenzhen, 6 hours historical data are used to predict 3 hours (horizon 6) and 6

hours (horizon 12) charging demand. In principle, any prediction model can be used. However, to reflect practical device constraints, we employ a two-layer LSTM+MLP model, with LSTM hidden dimensions of 256 and 128. For Fed-GAME, the learning rate is set as 0.0005, with $w_{self} = 0.6$, $\mu = 0.2$, $\alpha = \beta = 0.5$. Four shared scoring experts, and Top-$k = 2$ was used. Hyperparameters were determined via preliminary experiments. To evaluate probabilistic forecasting, $q = \{0.1, 0.5, 0.9\}$, ICP should approach 0.8, while QS and MIL should be minimized. All experiments are implemented in Python 3.10 on an NVIDIA A100 GPU.

3.1. Comparison Analysis

The forecasting performance results of Palo Alto and Shenzhen are shown in Table. 2. On the Palo Alto dataset, Fed-GAME achieves the best average QS and ICP scores. While our method does not achieve the best MIL performance as we upload only parameter differences rather than full parameters, it still surpasses the baselines in QS and ICP. This confirms that Fed-GAME's aggregation strategy can effectively harness client knowledge without direct data sharing. On the Shenzhen dataset, with 200 stations (47 unseen during training), Fed-GAME improves Step-6 performance by 54.23% (QS) and 43.87% (ICP) over No_FL. For Step-12, improvements are 45.14% and 68.93%, demonstrating strong generalization under non-IID and partial client participation settings. These results highlight its strength in long-term forecasting and confirm Fed-GAME's effectiveness in both accuracy and generalization across diverse deployment scenarios.

3.2. Performance of the GAME Aggregator

We evaluate the effectiveness of GAME aggregator against two strong graph-based baselines: GraphSAGE [21], which uses a fixed aggregation, and GAT [22] that learns attention weights between nodes. In this ablation, only the server-side aggregation module is replaced. The results are shown in Table. 3. The experimental results demonstrate the superiority of the GAME aggregator. On both datasets, Fed-GAME significantly outperforms both GraphSAGE and GAT on QS and MIL. This performance demonstrates the effectivenes of GAME. While GAT computes a single attention score between nodes, our GAME aggregator utilizes a set of scoring experts to evaluate client relationships from multiple perspectives. The personalized noisy Top-k router learns a sparse, adaptive strategy to combine these expert's judgment for each client. This approach allows GAME to learn effective content-aware aggregation function, resulting in better forecasting accuracy.

3.3. Communication Efficiency Evaluation

We compare the communication cost of Fed-GAME with baselines (FedAvg, FedProx and pFedMe). For the LSTM model ($\approx$ $996K$ parameters), in 6-step prediction (Palo Alto/Shenzhen) task, linear head = 2,322 parameters, the fraction of parameters selected for communication $r = \frac{|\theta_{\text{linear}}|}{|\theta|} = \frac{2{,}322}{996{,}013} \approx 0.0023$, which results in only 0.115% overhead. In 12-step prediction (Palo Alto/Shenzhen) task, linear head = 4,644 parameters, $r = \frac{4{,}644}{994{,}852} \approx 0.0047$, which increase 0.235% overhead. Thus, Fed-GAME achieves personalization at negligible cost compared to full-parameter baselines.

Table 2. Performance comparison of different FL forecasting models on Palo Alto and Shenzhen datasets

	Method	Aggregation Method	Palo Alto			Shenzhen		
			QS	MIL	ICP	QS	MIL	ICP
Step 6	No_FL	—	1.5228	4.7110	0.7585	18.6698	39.8802	0.6988
	LSTM-FedAvg	Average	1.4846	4.5517	0.7390	10.4788	10.1150	0.7381
	FedProx	Average	1.5378	4.7564	0.6571	9.8572	**6.3500**	0.6587
	pFedMe	Average	1.7047	5.0307	0.6677	10.5375	9.5557	0.7176
	PAG-FedAvg	Average	1.5304	**4.2462**	0.7534	13.7907	13.0350	0.4305
	GCRN-FedAvg	Average	1.4219	4.2518	0.7448	9.9119	9.7647	0.7002
	Fed-GAME	GAME	**1.4067**	4.4833	**0.7827**	**8.5461**	9.8635	**0.7432**
Step 12	No_FL	—	2.4770	7.0310	0.6726	19.1557	35.0342	0.6310
	LSTM-FedAvg	FedAvg	1.8118	5.4041	0.6633	18.4644	14.2590	0.7351
	FedProx	FedAvg	2.1077	5.4319	0.6662	11.8104	12.4952	0.7260
	pFedMe	FedAvg	2.1397	5.0307	0.6963	12.7760	13.5238	0.7077
	PAG-FedAvg	FedAvg	1.8843	5.0488	0.6866	12.7842	**13.3762**	0.6907
	GCRN-FedAvg	FedAvg	1.6297	5.1615	0.7277	9.3349	17.9791	0.7431
	Fed-GAME	GAME	**1.6270**	**5.0027**	**0.7569**	**10.5097**	15.5639	**0.7475**

Table 3. QS, MIL and ICP with different Aggregation Mechanism on different dataset (Prediction Step=12)

	Method	Aggregation Method	QS	MIL	ICP
Palo Alto	No_FL	—	2.477	7.031	0.6726
	Fed-GraphSAGE	GraphSAGE	1.8878	5.6907	0.756
	Fed-GAT	GAT	1.9081	5.6808	0.7462
	Fed-GAME	GAME	**1.627**	**5.0027**	**0.7569**
Shenzhen	No_FL	—	19.1557	35.0342	0.631
	Fed-GraphSAGE	GraphSAGE	11.9333	19.5901	**0.756**
	Fed-GAT	GAT	12.0329	19.3822	0.7392
	Fed-GAME	GAME	**10.5097**	**15.5639**	0.7475

3.4. Analysis of Learned Aggregation Weights

We further analyze the learned aggregation weights to better substantiate the proposed mechanism. The results show that the personalized weights w_{ij} are not solely determined by geographic proximity; instead, clients exhibit specialization across different temporal scales, leading to structured attention patterns. As training progresses, the aggregation weights evolve from near-uniform distributions to more differentiated profiles with higher variance and lower entropy, indicating the capture of non-trivial inter-client correlations under non-IID data distributions and yielding a 54.23% QS improvement on the Shenzhen dataset (Table. 2).

4. CONCLUSION

In this paper, we proposed a novel PFL framework, Fed-GAME, for time-seires forecasting by employing GAME aggregator to learn dynamic, content-aware graph attention coefficients. Experimental results demonstrate that FED-GAME achieves state-of-the-art predictive performance on heterogeneous time-series data. For future work, we plan to further investigate adaptive strategies for parameter selection in the selective update process.

5. REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, and etc, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, and etc, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[3] A. Z. Tan, H. Yu, and etc, "Towards personalized federated learning," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9587–9603, 2022.

[4] Z. Y. Wang, H. L. Xu, J. C. Liu, and etc, "Accelerating federated learning with cluster construction and hierarchical aggregation," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3805–3822, 2022.

[5] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in neural information processing systems*, vol. 33, pp. 21394–21405, 2020.

[6] Y. J Cheng, Z. W Zhang, and S. J. Wang, "Fedsds: Adaptive structured dynamic sparsity for federated learning under heterogeneous clients," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 9231–9235.

[7] F. W. Chen, G. D. Long, and etc, "Personalized federated learning with graph," *arXiv preprint arXiv:2203.00829*, 2022.

[8] C. Z. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 1202–1211.

[9] R. Liu, P. W. Xing, Z. C. Deng, and etc, "Federated graph neural networks: Overview, techniques, and challenges," *IEEE transactions on neural networks and learning systems*, 2024.

[10] Z. R. Zhou, G. Y. Gao, X. H. Wu, and etc., "Personalized federated learning via learning dynamic graphs," *arXiv preprint arXiv:2503.05474*, 2025.

[11] R. Ye, Z. Y. Ni, F. Z. Wu, and etc., "Personalized federated learning with inferred collaboration graphs," 2023, vol. 202 of *Proceedings of Machine Learning Research*, pp. 39801–39817, PMLR.

[12] Y. Li, R. Y. Xie, C. J. Li, and etc., "Federated graph learning for ev charging demand forecasting with personalization against cyberattacks," *arXiv preprint arXiv:2405.00742*, 2024.

[13] X. H. Wu, X. F. Liu, J. W. Niu, and etc., "Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 19375–19384.

[14] X. H. Wu, X. F. Liu, and J. W. Niu, "Decoupling general and personalized knowledge in federated learning via additive and low-rank decomposition," 2024, MM '24, p. 7172–7181.

[15] Y. Li, Fan. M. F, G. Chen, C. J Li, and Biplab Sikdar, "Zero-shot goose anomaly detection via multigate mixture-of-experts with pre-trained large language model," in *2025 IEEE Kiel PowerTech*, 2025, pp. 1–6.

[16] City of Palo Alto, "Electric vehicle charging station usage. open data. city of palo alto, 2021.," `https://data.cityofpaloalto.org/dataviews/257812/electric-vehicle-charging-station-usage-july-2011-dec-2020/`.

[17] J. Zhou and L. Ma, "Analysis on the evolution characteristics of shenzhen residents' travel structure and the enlightenment of public transport development policy," *Urban Mass Transit*, vol. 24, no. 7, pp. 63–68, 2021.

[18] H. H. Qu, H. X. Kuang, and etc., "A physics-informed and attention-based graph learning approach for regional electric vehicle charging demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 14284–14297, 2024.

[19] S. R. Fahim, R. Atat, and etc., "Forecasting ev charging demand: a graph convolutional neural network-based approach," in *2024 4th International Conference on Smart Grid and Renewable Energy (SGRE)*. IEEE, 2024, pp. 1–6.

[20] Y. Li, G. Chen, and X. J. Zhou, "A multiscale time-aware graph mamba neural network-based method for multienergy system consumption quantiles forecasting," *IEEE Systems Journal*, pp. 1–11, 2025.

[21] W. Hamilton, Z. T. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*. 2017, vol. 30, Curran Associates, Inc.

[22] Petar V., Guillem C., Arantxa C., and etc., "Graph attention networks," in *6th International Conference on Learning Representations, 2018, Conference Track Proceedings*, 2018.