Federated Prompt Learning for Weather Foundation Models on Devices

Shengchao Chen, **Guodong Long**, **Tao Shen**, **Jing Jiang** and **Chengqi Zhang**
Australian Artificial Intelligence Institute, FEIT, University of Technology Sydney
shengchao.chen.uts@gmail.com, {guodong.long, tao.shen, jing.jiang, chengqi.zhang}@uts.edu.au

Abstract

On-device intelligence for weather forecasting uses local deep learning models to analyze weather patterns without centralized cloud computing, holds significance for supporting human activates. Federated Learning is a promising solution for such forecasting by enabling collaborative model training without sharing raw data. However, it faces three main challenges that hinder its reliability: **(1)** data heterogeneity among devices due to geographic differences; **(2)** data homogeneity within individual devices and **(3)** communication overload from sending large model parameters for collaboration. To address these challenges, this paper propose **Fed**erated **P**rompt learning for Weather Foundation Models **o**n **D**evices (**`FedPoD`**), which enables devices to obtain highly customized models while maintaining communication efficiency. Concretely, our *Adaptive Prompt Tuning* leverages lightweight prompts guide frozen foundation model to generate more precise predictions, also conducts prompt-based multi-level communication to encourage multi-source knowledge fusion and regulate optimization. Additionally, *Dynamic Graph Modeling* constructs graphs from prompts, prioritizing collaborative training among devices with similar data distributions to against heterogeneity. Extensive experiments demonstrates **`FedPoD`** leads the performance among state-of-the-art baselines across various setting in real-world on-device weather forecasting datasets.

1 Introduction

Climate change has a profound impact on both natural ecosystems and human societies [Karl *et al.*, 2009; Kjellstrom *et al.*, 2016]. It leads to higher temperatures, sea level changes and more frequent extreme weather events [Hagemann *et al.*, 2013]. As a result, precise weather forecasting is becoming increasingly important. Data from meteorological devices in various regions is vital. However, analyzing this data with deep learning through centralized cloud computing presents challenges such as network dependence and privacy concerns [Chakraborty and Rodrigues, 2020]. First, sending large volumes of data to centralized system places a heavy burden on communication networks, which is impractical for low-resource weather devices. Second, data from sensitive locations is subject to privacy laws, restricting sharing across devices [Chen *et al.*, 2023a]. To address these issues, on-device intelligence for analyzing data directly on the devices is crucial, reduces the need for data transfers, protects privacy, and decreases reliance on networks.

Federated Learning (FL) [McMahan *et al.*, 2017] is a promising method for on-device intelligence that trains a uniform model collaboratively across multiple devices without exchanging data. However, the models often underperform due to statistical heterogeneity among clients and data homogeneity on within individual clients' data. Personalized FL (PFL) offers new insights by developing specialized models for each device, enabling tailored on-device intelligence [Paulik *et al.*, 2021]. Recent PFL methods have introduced various methods to improve personalization [Chen *et al.*, 2022; Tan *et al.*, 2022; Li *et al.*, 2021b]. Despite these advances, two significant challenges remain. First, there is often inadequate consideration of the impact of physical geographic location on local models. For example, devices on seashores and hilltops may collect different data types even if they are geographically close. Second, the substantial communication demands of large neural networks burden both clients and servers. Edge devices with limited resources may struggle to process the necessary updates for these complex models [Xiong *et al.*, 2023]. Moreover, the transfer of entire model parameters hampers communication efficiency.

To tackle the above issues, this paper introduces **Fed**erated **P**rompt Learning for Weather Foundation Models **o**n **D**evices (**`FedPoD`**), which allows devices to obtains high customized models with efficient communication. **`FedPoD`** comprising two pivotal components: (1) Adaptive Prompt Tuning and (2) Dynamic Graph Modeling. Adaptive Prompt Tuning against data homogeneity and reduces communication load via updating local prompts based on the frozen foundation model (FM) to capture local information and guide FM to generate accurate prediction, coupled with multi-level communication. Additionally, **`FedPoD`** uses Dynamic Graph Modeling on the server to manage prompts from clients and to build multiple graphs dynamically, considering various perspectives. This process takes geographic features into account and promotes priority collaborative learning among clients with similar data, mitigating the effects of data heterogeneity. As shown in Table 1, using a pre-trained foundation model leads to fewer parameters and higher performance compared to starting from scratch with FedAvg [McMahan *et al.*, 2017]. Furthermore, **FedPoD** achieves the best results with the proposed adaptive prompt tuning and dynamic graph modeling.

Method	Trainable Param.	MAE/RMSE
Train from scratch (FedAvg)	5,284,173	40.3/51.2
Pre-trained FM (FedAvg)	215,089	33.5/44.5
Pre-trained FM & Prompts (FedAvg)	**159,649**	31.1/41.9
FedPoD (Ours)	**159,649**	**27.0/37.6**

Table 1: Compared with training Encoder-only Transformer as the foundation model. Experiments are implemented with FedAvg, and our method. Communication rounds: 30, local updating: 5.

Main Contributions. With extensive experiments across datasets including real-world on-deivce weather series datasets on various setting, we show that our **FedPoD** consistently outperforms state-of-the-art baselines. Besides, we conduct further analysis to provide more insights in **FedPoD** from the perspective of ablation, hyperparameter sensitivity, and privacy. The main contributions is as follows:

- We present **FedPoD**, a communication-efficient framework for on-device weather forecasting that addresses the challenges of data heterogeneity among devices and data homogeneity within individual clients during federated learning.
- We show *Adaptive Prompt Tuning* that uses prompts to represent information and guide generation. These prompts enable multi-level communication and knowledge sharing, reducing the impact of data homogeneity.
- We introduce *Dynamic Graph Modeling* to create dynamic links between participants' prompts. This prioritizes collaborative optimization for clients with similar representations, enhancing personalization.
- With extensive experiments, we show **FedPoD** consistently achieves the best and help improve the communication efficiency while keeping privacy, and adaptive prompt tuning also benefits baselines.

2 Related Work

Weather Forecasting. Weather forecasting is a crucial tool that analyzes the variations in weather patterns. Recently, weather forecasting has made significant strides by incorporating data-driven approaches [Chen and Lai, 2011; Sapankevych and Sankar, 2009; Voyant *et al.*, 2012]. RNNs have shown promising in weather forecasting [Shi *et al.*, 2015; Grover *et al.*, 2015]. Besides, Transformers [Zhou *et al.*, 2021; Zhou *et al.*, 2022; Wu *et al.*, 2021; Chen *et al.*, 2023c] can capture non-stationary changes, which have contributed to their widespread use in weather analysis. To overcome challenges caused by intricate spatial-temporal correlation, spatial-temporal modeling methods [Yu *et al.*, 2017] can be an effective solution. However, these methods all focus on data-intensive centralized training, which poses a challenge to weather forecasting practices.

Personalized Federated Learning. Weather forecasting involves significant communication loads and raises privacy issues due to the large volume of data processes on parallel [Chavan and Momin, 2017]. Federated learning (FL) [McMahan *et al.*, 2017] offers a way to perform on-device intelligence but is often hampered by data heterogeneity and homogeneity. Personalized FL (PFL) seeks to overcome these issues by training customized models for each device, providing fresh insights. For example, [T Dinh *et al.*, 2020; Hanzely *et al.*, 2020; Li *et al.*, 2021a] add a regularization that decomposes the personalized model optimization from the global. [Li *et al.*, 2021b; Collins *et al.*, 2021] share part of the model and keep personalized layers private. [Zhang *et al.*, 2020] enables a flexible method by adaptively weighted aggregation. [Fallah *et al.*, 2020] start from a Model-Agnostic Meta-Learning where a meta-model is learned to generate the initialized local model for each client. In addition, [Chen *et al.*, 2022] utilize structure information to explore the topological relations among clients.

3 Preliminaries and Problem Formulation

Weather Forecasting. A multivariate weather time series represented by $\boldsymbol{X}_i \in \mathbb{R}^{m\times n}$, where m and n is the series length and the number of variables, respectively. Each data point is shown as $\boldsymbol{x}_t \in \mathbb{R}^{1\times n}$. The weather forecasting task can be divided into two categories below:

- **Task 1-Multivariate to Univariate Forecasting**: Predicting a single variable for future Q periods using all variables from the past P periods.
- **Task 2-Multivariate to Multivariate Forecasting**: Predicting all variables for future Q periods from all variables in the past P periods.

These tasks can be defined as follows:

$$\begin{aligned}
&\textbf{Task1:}\quad [\boldsymbol{x}_{t-P}, \boldsymbol{x}_{t-P+1}, \cdots, \boldsymbol{x}_t] \xrightarrow{f} \left[\boldsymbol{x}_{t+1}^{T1}, \boldsymbol{x}_{t+2}^{T1}, \cdots, \boldsymbol{x}_{t+Q}^{T1}\right],\\
&\textbf{Task2:}\quad [\boldsymbol{x}_{t-P}, \boldsymbol{x}_{t-P+1}, \cdots, \boldsymbol{x}_t] \xrightarrow{f} \left[\boldsymbol{x}_{t+1}^{T2}, \boldsymbol{x}_{t+2}^{T2}, \cdots, \boldsymbol{x}_{t+Q}^{T2}\right],
\end{aligned} \tag{1}$$

where f denotes the learning system, $\boldsymbol{x}_t^{T1} \in \mathbb{R}^{1\times 1}$ is the predicted variable at the t-th step, and $\boldsymbol{x}_t^{T2} \in \mathbb{R}^{1\times n}$ is the predicted variable at the t-th step.

On-device Weather Forecasting based on FL. Each device[1] possesses a local data varying location pattern, leading to statistical heterogeneity. Thus, we can define the task on-device weather forecasting as:

$$[f_1(D_1), f_2(D_2), ..., f_N(D_N)] \rightarrow [D'_1, D'_2, ..., D'_N] \tag{2}$$

where the D_k and D'_k denote the input dataset and prediction in k-th client, respectively, and f_k is the personalized model for k-th client. This makes vanilla FL that train an uniform model unsuitable, and the task is updated to the PFL problem that solves below bi-level optimization.

$$\begin{aligned}
F(v; w)\colon = &\operatorname*{arg\,min}_{\{v_1, v_2, ..., v_N\}} \sum_{k=1}^{N} \frac{n_k}{n} F_k(v_k) + \lambda \mathcal{R}(v_k, w),\\
s.t. \quad & w \in \operatorname*{arg\,min}_{w} G(F_1(w), F_2(w), ..., F_N(w)),
\end{aligned} \tag{3}$$

[1]We take "device(s)" and "client(s)" to mean the same one.

where each client hold a customized model parameterized by v_i, w denotes the global model. $\mathcal{R}(\cdot)$ is a regularization term, $G(\cdot)$ is the aggregation strategy. Previous studies have had difficulty managing the non.iid of geographic data, often overlooking how spatial-temporal correlation is affected by more than just location [Chen *et al.*, 2023b]. In this work, we aim to address two main challenges: **(1)** *How can we ensure efficient communication between clients and servers while guaranteeing the framework's high performance?* **(2)** *How can we minimize the heterogeneity caused by complex geographic features in the most cost-effective way?*

4 Methodology

In this section, we detail our **`FedPoD`**, illustrated in Fig. 1. Each client hold a pre-trained FM (PFM)[2] and three types of prompts that updated locally. In each round, we introduce multi-level communication based on prompts uploaded by participants, including inter-clients and client-server. In the server, we present a novel aggregation method, Dynamic Graph Modeling, to building dynamic graphs based on structural information from prompts, reducing influence of data heterogeneity. With the updated prompts from the server, clients perform local optimization with a specialized prompt-wise loss. We'll describe them in more detail below.

Adaptive Prompt Tuning. We introduce Adaptive Prompts Tuning for local updating to minimize the effects of data homogeneity within devices while keeping computational loads low. Unlike prompt tuning in Natural Language Processing, which simply adjusts inputs to guide a pre-trained large language model (LLM) to produce outputs [White *et al.*, 2023]. It involves using lightweight prompts that dynamically represent local knowledge and act as information carriers in multi-level communication. This helps lessen the overall impact of both global data heterogeneity and local data homogeneity during collaborative training. Specifically, we use trainable parameters as prompts, including TEMPORAL PROMPTS ($\boldsymbol{P}_T$) and INTER-VARIABLES PROMPTS ($\boldsymbol{P}_V$), to capture the local temporal dynamics and the relationships among variables. These prompts are incorporated into the time series and are refined during the local training phase. The updating process of $\boldsymbol{P}_T$ and $\boldsymbol{P}_V$ is shown in Alg. 1.

After updating the TEMPORAL PROMPTS ($\boldsymbol{P}_T$) and INTER-VARIABLES PROMPTS ($\boldsymbol{P}_V$), we apply two learnable matrices, $\boldsymbol{W}_t$ and $\boldsymbol{W}_v$, to them. This is represented as $\boldsymbol{X} = \boldsymbol{P}_T \odot \boldsymbol{W}_t + \boldsymbol{P}_V \odot \boldsymbol{W}_v$. These matrices help to adjust the significance of the prompts, ensuring they contribute to our optimization goal without straying off course. Furthermore, we introduce SPATIAL PROMPT ($\boldsymbol{P}_S$), to encode local geographic pattern for comprehensive modeling, via updating with original input $\boldsymbol{X}_{ipt}$ and $\boldsymbol{X}$. The final prediction $\overline{\boldsymbol{X}}$ is then derived using the following formula:

$$\begin{aligned}\boldsymbol{P}_S, \boldsymbol{X} \leftarrow & \texttt{LayerNorm}(\|\boldsymbol{X}_{\text{ipt}}, \boldsymbol{X}_{\text{geo}}\|, \|\boldsymbol{X}, \boldsymbol{P}_S\|), \\ \overline{\boldsymbol{X}} = & \texttt{FFN}(\boldsymbol{F}(\boldsymbol{X}_{\text{ipt}} + \boldsymbol{X}))\end{aligned} \tag{4}$$

[2]Detailed information about the utilized PFM in Appendix B. Complete version: *https://arxiv.org/abs/2305.14244*.

Algorithm 1 Implementation of $\boldsymbol{P}_T$ and $\boldsymbol{P}_V$ Updating

Initialize Original input series $\boldsymbol{X}_{\text{ipt}}$, frozen PFM $\boldsymbol{F}_M$, Temporal/Variable updating steps K_t and K_s.
for time forecasting step $q = 1, 2, \ldots$ **do**
  $\texttt{Updating}(\boldsymbol{F}_M(\|\boldsymbol{X}_{\text{ipt}}, \boldsymbol{P}_T\|^T)), \boldsymbol{P}_T \in \mathbb{R}^{q \cdot K_t \times n}$
  ▷ $\|.\|^T$: *concat* along temporal dimension
  $\boldsymbol{P}_T \leftarrow \|\boldsymbol{P}_T, \boldsymbol{P}'_T \in \mathbb{R}^{K_t \times n}\|^T$
  ▷ $\boldsymbol{P}'_T$: Next temporal prompt block
end for
for variable forecasting step $p = 1, 2, \ldots$ **do**
  $\texttt{Updating}(\boldsymbol{F}_M(\|\boldsymbol{X}_{\text{ipt}}, \boldsymbol{P}_V\|^V)), \boldsymbol{P}_V \in \mathbb{R}^{m \times p \cdot K_v}$
  ▷ $\|.\|^V$: *concat* along variable dimension
  $\boldsymbol{P}_V \leftarrow \|\boldsymbol{P}_V, \boldsymbol{P}'_V \in \mathbb{R}^{m \times K_v}\|^V$
  ▷ $\boldsymbol{P}'_V$: Next inter-variable prompt block
end for

where $\boldsymbol{X}_{\text{geo}}$ denotes the client's geographic location represented by (ϕ, λ), ϕ and λ is the latitude and longitude coordinates, respectively, for simultaneous updating of $\boldsymbol{P}_S$ and $\boldsymbol{X}$ to adjust the parameters of $\boldsymbol{P}_S$.

Local Optimization from Multi-Task Perspective. For each client's local optimization, we focus on two key elements: (1) multi-level communication regularization and (2) a multi-task perspective. The first involves interactions among clients and between clients and the server, aiming to mitigate the effects of data homogeneity. The second treats the optimization of various prompts as separate tasks, helping to lessen the unpredictability that comes with mixed updates. Consequently, we suggest a prompt-based optimization objective for local updates, as follows:

$$\mathcal{L}_{ap} = \texttt{MSE}(y', y) + \mathcal{R}(\{\boldsymbol{P}_i\}; \{\boldsymbol{P}_j\}^l; \{\boldsymbol{P}_i\}^l; \{\boldsymbol{P}\}^*), \tag{5}$$

where $\texttt{MSE}(\cdot)$ denotes the mean square error loss that evaluate the distance between ground-truth y and output y', $\mathcal{R}(\{\boldsymbol{P}_i\}; \{\boldsymbol{P}_j\}^l; \{\boldsymbol{P}_i\}^l; \{\boldsymbol{P}\}^*)$ is the regularization term utilized to measure the distance between prompts, including personalized prompts $\{\boldsymbol{P}_i\}^l$ of the i-th client, neighboring j-th client's prompts $\{\boldsymbol{P}_j\}^l$, and global prompts $\{\boldsymbol{P}\}^*$ obtained by averaging all client's prompts. The underlying motivations is allowing local client to craft highly customized models via decomposing prompt parameters from the neighboring and global prompts while keeping the comprehensive knowledge. Then, inspired by [Kendall *et al.*, 2018], we conceptualize the optimization from a multi-task view, can be formulated as:

$$\begin{aligned}\mathcal{L}_{ap} = & \texttt{MSE}(y', y) \\ & + \frac{1}{\xi^2} L^2(\{\boldsymbol{P}_i\}, \{\boldsymbol{P}\}^*) + \frac{1}{\xi^2} L^2(\{\boldsymbol{P}_i\}, \{\boldsymbol{P}_i\}^l) \\ & + \frac{1}{\tau^2} \cdot \frac{1}{(|\mathcal{N}|/S_{\boldsymbol{G}}) - 1} \sum_{j \in \mathcal{N}} L^2(\{\boldsymbol{P}_i\}, \{\boldsymbol{P}_j\}^l) \\ & + 4\{\log_2(\xi) + \log_2(\tau)\}.\end{aligned} \tag{6}$$

Here, the ξ and τ are importance coefficients that obey $\lambda, \tau \in (0, 1)$, the L^2 is L2 regularization (e.g., Euclidean distance, Cosine Similarity, etc.). $S_{\boldsymbol{G}}$ represents the subgraph step used to adjust the scope of interaction between clients. The inter-client regularization term $\frac{1}{\tau^2} \cdot \frac{1}{(|\mathcal{N}|/S_{\boldsymbol{G}})-1} \sum_{j \in \mathcal{N}} L^2(\{\boldsymbol{P}_i\}, \{\boldsymbol{P}_j\})$, drives the local up-

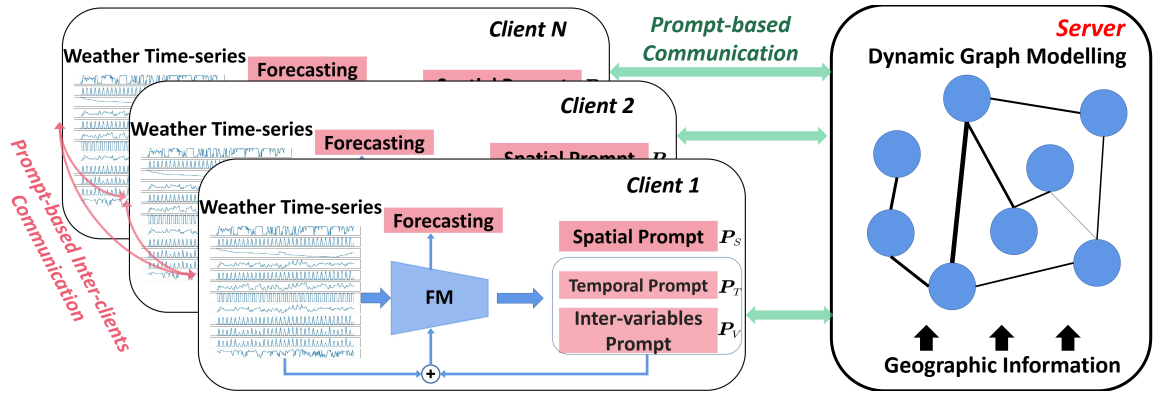

Figure 1: Architecture of **FedPoD**, prompts comprise the Spatial Prompt, Temporal Prompt, and Inter-variables Prompt. $\leftrightarrow$: communication exchanges prompts among clients, $\leftrightarrow$: communication between clients and the server only transmit prompts.

dating process towards a more comprehensive representation via considering neighboring clients with distinct feature distributions within a given range. Regulation terms $\frac{1}{\xi^2}L^2(\{\boldsymbol{P}_i\},\{\boldsymbol{P}\}^*)$ and $\frac{1}{\xi^2}L^2(\{\boldsymbol{P}_i\},\{\boldsymbol{P}_i\}^l)$ are employed with the purpose of prompting the local clients to attain a more personalized representation.

Dynamic Graph Modeling for Global Aggregation. We introduce Dynamic Graph Modeing (DGM) on the server to boost personalization by constructing spatial-temporal correlations among clients. This promotes collaborative optimization among clients with similar local knowledge representation. DGM uses the prompts shared by clients and their geographic data, like latitude and longitude, to form graphs. These graphs reveal possible relationships among clients, leading to a more customized optimization process. Specifically, we divide local prompts into three classes: (1) Temporal and Inter-Variables Prompts $\{\boldsymbol{P}_{T,i},\boldsymbol{P}_{V,i}\}_{i=1}^N$; (2) Spatial Prompts $\{\boldsymbol{P}_{S,i}\}_{i=1}^N$ and (3) Full Prompts $\{\boldsymbol{P}_i\}_{i=1}^N$, where N is the number of clients. First, the server generates a static graph $\boldsymbol{A}_{\text{geo}}$ according to the location information based on Haversine formula [Robusto, 1957] as:

$$2R\cdot\tan^{-1}\left(\sqrt{\frac{\sin^2(\frac{\Delta\phi}{2})+\cos(\phi_i)\cdot\cos(\phi_j)\cdot\sin^2(\frac{\Delta\lambda}{2})}{1-(\sin^2(\frac{\Delta\phi}{2})+\cos(\phi_i)\cdot\cos(\phi_j)\cdot\sin^2(\frac{\Delta\lambda}{2}))}}\right), \tag{7}$$

where $i,j\in\mathcal{N}, i\neq j$, ϕ_i and ϕ_j are the latitude coordinates of client i and j, respectively, $\Delta\phi=\phi_i-\phi_j$ is the difference in latitude between the two points in radians, $\Delta\lambda=\lambda_i-\lambda_j$ is the difference in longitude between the client i and client j, R is the radius of the Earth.

To grasp the potential correlations between clients dynamically, we use two matrices, $\boldsymbol{W}_i$ are $\boldsymbol{W}_j$, to apply linear transformations to the prompt vectors $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ of two different clients. The relation of the i-th client to the j-th client is calculated using the formula $e_{i,j}=\alpha(\boldsymbol{W}_i\boldsymbol{P}_i,\boldsymbol{W}_j\boldsymbol{P}_j)$, where $\alpha(\cdot)$ denotes a shared attention mechanism that operates in the space $\mathbb{R}^{F'}\times\mathbb{R}^{F'}\rightarrow\mathbb{R}$. Here, $\mathbf{W}\in\mathbb{R}^{F'\times F}$ helps determine the attention coefficients. We then introduce another matrix $\boldsymbol{W}$ to calculate the weight of the connection (edge) and construct an adjacency matrix as follows:

$$\boldsymbol{A}_{i,j}=\frac{e_{i,j}}{1+e^{-\boldsymbol{W}[\boldsymbol{W}_i\boldsymbol{P}_i-\boldsymbol{W}_j\boldsymbol{P}_j]}}. \tag{8}$$

For three types of prompts, we create three corresponding adjacency matrices (graphs), denoted as $\boldsymbol{A}_{TV}$, $\boldsymbol{A}_S$, and $\boldsymbol{A}$, via Eq. 8. We then merge these with $\boldsymbol{A}_{Geo}$ (from Eq.7) using an attention mechanism to capture more precise correlation representations. Based on these matrices, we reconstruct prompts to deliver personalized prompts $\{P_i\}^l$ for each client:

$$\begin{aligned}\boldsymbol{A}'&\leftarrow\texttt{Softmax}\left(\frac{(\boldsymbol{A}_{Geo}-\boldsymbol{A}_S)\boldsymbol{A}_{TV}^\top}{\sqrt{d_k}}\right)\boldsymbol{A},\\ \{P_i\}_{i=1}^{l,N}&\leftarrow\alpha\boldsymbol{A}\{P_i\}_{i=1}^N+(1-\alpha)\boldsymbol{A}'\{P_i\}_{i=1}^N,\end{aligned} \tag{9}$$

where $\sqrt{d_k}$ is the dimension of adjacent matrix, and α is importance coefficient. The term $[\boldsymbol{A}_{\text{Geo}}-\boldsymbol{A}_{\text{S}}]$ highlights the discrepancy between the actual geographic correlation and the encoded spatial correlation, enabling the dynamic adjustment of spatial-temporal correlation among clients to achieve a more precise potential correlation graph modeling.

Optimization for FedPoD. The overall optimization objective of **FedPoD** is to solve a bi-level optimization problem, as below:

$$\begin{aligned}&\underset{\{\boldsymbol{P}_i\};\boldsymbol{A}}{\arg\min}\ \sum_{i=1}^N[\frac{n_i}{n}F_i(\{\boldsymbol{P}_i\};D_i)+\mathcal{R}(\{\boldsymbol{P}_i\};\{\boldsymbol{P}_j\}^l;\{\boldsymbol{P}_i\}^l;\{\boldsymbol{P}\}^*)]\\&+\tau\mathcal{G}(\boldsymbol{A}),\\ &s.t.\quad \{\boldsymbol{P}\}^*\in\underset{\{\boldsymbol{P}_1\},\ldots,\{\boldsymbol{P}_N\}}{\arg\min}\sum_{i=1}^N\frac{n_i}{n}F_i(\{\boldsymbol{P}_i\}),\\ &\qquad\{\boldsymbol{P}\}^l\in\underset{\{\boldsymbol{P}_i\}^l}{\arg\min}\sum_{j\in\mathcal{N}}\boldsymbol{A}_{j,i}S(\{\boldsymbol{P}_i\}^l,\{\boldsymbol{P}_j\}^l)\end{aligned} \tag{10}$$

where $\{\boldsymbol{P}\}$ denotes local prompts including $\boldsymbol{P}_T$, $\boldsymbol{P}_V$, and $\boldsymbol{P}_S$, $\{\boldsymbol{P}\}^*$ is global prompts, the local model was parameterized by $\{\boldsymbol{P}\}$ after receiving the pre-trained FM. The $\{\boldsymbol{P}_j\}^l$ is personalized local models from other clients that achieve by the additional regularization term $\mathcal{G}(\cdot)$ that is a graph-based constraint that ensures each client aggregates with similar neighbor nodes. The learned graph with the adjacent matrix $\boldsymbol{A}$

Algorithm 2 Implementation of **`FedPoD`**

Initialize local data $\{D_i\}_{i=1}^N$, foundation model $\boldsymbol{F}_M$, prompts $\{\boldsymbol{P}_{T,i}, \boldsymbol{P}_{V,i}, \boldsymbol{P}_{S,i}\}_{i=1}^N$
Initialize $\{\boldsymbol{P}_{T,i}, \boldsymbol{P}_{V,i}, \boldsymbol{P}_{S,i}\}_{i=1}^N$ as $\{\boldsymbol{P}_i\}_{i=1}^N$, $\boldsymbol{W}_{bt,i}$, $\boldsymbol{W}_{bs,i}$
Server-side: Broadcast frozen $\boldsymbol{F}_M$ to each clients ▷ Model communication
for rounds $R = 1, 2, 3...$ **do** ▷ FL rounds in sequence
  Client-side:
  Download $\{\boldsymbol{P}\}^l$ (*personalized prompts*), $\{\boldsymbol{P}\}^*$ (*global prompts*) from the server
  for each client i in parallel **do** ▷ Clients in parallel
    $\{\boldsymbol{P}_i\} \leftarrow$ LocalUpdate$(\boldsymbol{F}_M, D_i, \{\boldsymbol{P}_i\}_{i=1}^N)$ ▷ Prompt-based training
    Upload $\{\boldsymbol{P}_i\}$ to the server ▷ Model communication
  end for
  Server-side:
  $\boldsymbol{A}_{\text{geo}} \leftarrow$ Haversine Formula(ϕ, λ) (*Eq. 7*) ▷ Generate the static graph
  $\boldsymbol{A} \leftarrow$ Dynamic Graph Modeling$(\{\boldsymbol{P}_T\}_{i=1}^N, \{\boldsymbol{P}_V\}_{i=1}^N, \{\boldsymbol{P}_S\}_{i=1}^N)$ (*Eq. 8*) ▷ Generate the dynamic graph
  $\boldsymbol{A}_{\text{TV}} \leftarrow$ Dynamic Graph Modeling$(\{\boldsymbol{P}_T\}_{i=1}^N, \{\boldsymbol{P}_V\}_{i=1}^N)$ (*Eq. 8*) ▷ Generate the dynamic graph
  $\boldsymbol{A}_{\text{S}} \leftarrow$ Dynamic Graph Modeling$(\{\boldsymbol{P}_T\}_{i=1}^N, \{\boldsymbol{P}_V\}_{i=1}^N)$ (*Eq. 8*) ▷ Generate the dynamic graph
  $\boldsymbol{A}' \leftarrow$ Attention$(\boldsymbol{A}, \boldsymbol{A}_{\text{TV}}, \boldsymbol{A}_{\text{S}}, \boldsymbol{A}_{\text{geo}})$ (*Eq. 9*) ▷ Attention for filtering
  $\{\boldsymbol{P}_i\}_{i=1}^{l,N} \leftarrow \alpha \boldsymbol{A}\{\boldsymbol{P}_i\}_{i=1}^N + (1-\alpha)\boldsymbol{A}'\{\boldsymbol{P}_i\}_{i=1}^N$ (*According to Eq. 9*) ▷ Update personalized prompts
  $\{\boldsymbol{P}_i\}^* \leftarrow \frac{n}{n_k}\sum_{i=1}^N \boldsymbol{P}^s$, $w_r \leftarrow \frac{n}{n_k}\sum_{i=1}^N w_{r,i}$ ▷ Update global prompts and layers
end for
LocalUpdate$(\boldsymbol{F}_M, D, \boldsymbol{P}_{\text{T}}, \boldsymbol{P}_{\text{V}}, \boldsymbol{P}_{\text{S}}, \boldsymbol{F}_{layer})$
for local epoch $e = 1, 2, ...$ **do**
  Update $\boldsymbol{P}_{\text{T}}, \boldsymbol{P}_{\text{V}}$ (*Algorithm 1*)
  Update $\boldsymbol{P}_{\text{S}}$ (*Eq. 4*)
  Update rest of trainable parameters (*Eq. 4*)
  Compute local loss (*Eq. 6*) ▷ Optimization from multi-task view
end for

(computed by $\boldsymbol{A}', \boldsymbol{A}$) is expected to be sparse and able to preserve proximity relationships among clients. Algorithm 2 shows the detailed implementation of our **`FedPoD`**.

5 Theorems and Proofs

Theorem 1. *Consider a on-device weather forecasting system with m clients. Let $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m$ be the true data distribution and $\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, ..., \hat{\mathcal{D}}_m$ be the empirical data distribution. Denote the head h as the hypothesis from $\mathcal{H}$ and d be the VC-dimension of $\mathcal{H}$. The total number of samples over all clients is N. Then with probability at least $1-\delta$:*

$$\max_{(\{\boldsymbol{P}_1\},\{\boldsymbol{P}_2\},...,\{\boldsymbol{P}_m\})} \left| \sum_{i=1}^m \frac{|D_i|}{N} \mathcal{L}_{ap,\mathcal{D}_i} - \sum_{i=1}^m \frac{|D_i|}{N} \mathcal{L}_{ap,\hat{\mathcal{D}}_i} \right| \leq \sqrt{\frac{N}{2}\log\frac{(m+1)|\{\boldsymbol{P}\}|}{\delta}} + \sqrt{\frac{d}{N}\log\frac{eN}{d}} \tag{11}$$

Theorem 2 (Transmitting Prompts Ensure Privacy)**.** *Consider a device with a frozen pre-trained foundation model parameterized by θ_f, and trainable prompts parameterized by θ_p but initialized before updates. Transmitting these prompts can ensure privacy in multi-level communication.*

Proof. Detailed proofs can be found at Appendix C. □

6 Experiments

Datasets. Three weather multivariate time-series datasets from [Chen *et al.*, 2023b], including AvePRE, SurTEMP, and SurUPS collected by 88, 525, and 238 devices, respectively. Detailed information can be found at Appendix A.

Baselines. We compare with competitive FL methods, including FedAvg [McMahan *et al.*, 2017], FedProx [Li *et al.*, 2020], pFedMe [T Dinh *et al.*, 2020], Per-FedAvg [Fallah *et al.*, 2020], FedATT [Jiang *et al.*, 2020], APFL [Deng *et al.*, 2020], FedAMP [Huang *et al.*, 2021], and SFL [Chen *et al.*, 2022], while keeping the local foundation model consistent. Details about baselines can be found at Appendix A. In addition, we adapt two fine-tuning methods for each baseline for evaluate our method's effectiveness, as below:

- **Conventional Fine-tuning**: Update local FM with an FFN as the fine-tune head.
- **Adaptive Prompts Tuning (Ours)**: Update prompts with the frozen FM and multi-level communication.
- **Other Prompt Tuning**: Add parameters to input to updating models [Chen *et al.*, 2023b; Guo *et al.*, 2023].

Implementation. The task of on-device weather forecasting is to predict the next 12 hours using the data from the previous 12 hours. Main experiments are conducted in 25 local epoch within 50 communication round. Following [Chen *et al.*, 2022], Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used as evaluation metrics. All results are in $100\times$ the original value for a clearer comparison. Detailed information about the implementation, local updating process and the aggregation can be found at Appendix B.

Fine-Tuning Strategy	Method	AvePRE		SurTEMP		SurUPS	
		Task1	**Task2**	**Task1**	**Task2**	**Task1**	**Task2**
Conventional Fine-tuning	FedAvg [McMahan *et al.*, 2017]	34.6/44.8	56.0/90.1	47.6/64.4	56.5/78.3	53.5/74.2	54.1/74.6
	FedProx [Li *et al.*, 2020]	31.7/42.1	54.4/87.2	44.4/62.7	52.9/76.4	51.2/69.5	52.3/72.4
	Per-FedAvg [Fallah *et al.*, 2020]	30.9/40.7	54.3/71.5	41.4/60.9	51.8/73.3	50.2/69.7	51.7/71.8
	APFL [Deng *et al.*, 2020]	32.5/43.8	56.1/84.9	46.2/63.1	59.4/77.3	54.3/73.7	53.8/73.4
	FedAMP [Huang *et al.*, 2021]	31.9/41.3	54.7/84.2	43.8/62.9	52.3/73.7	51.5/70.0	53.2/73.4
	FedATT [Jiang *et al.*, 2020]	34.5/44.7	63.2/89.8	48.7/63.1	61.0/79.4	58.8/73.6	64.6/82/0
	pFedMe [T Dinh *et al.*, 2020]	32.2/42.7	64.0/85.2	42.9/61.8	50.7/74.6	51.7/70.1	52.5/72.0
	SFL [Chen *et al.*, 2022]	30.0/40.2	53.1/81.2	39.9/62.6	51.7/76.1	48.0/69.1	51.0/70.4
Adaptive Prompt Tuning (Ours)	FedAvg [McMahan *et al.*, 2017]	32.4/42.8	51.0/76.3	41.2/61.7	54.4/76.8	52.1/72.2	53.2/73.8
	FedProx [Li *et al.*, 2020]	27.1/38.0	47.1/70.2	39.7/61.5	51.7/75.2	48.1/67.1	51.0/67.6
	Per-FedAvg [Fallah *et al.*, 2020]	29.3/37.9	45.3/67.4	37.8/60.0	51.3/72.2	47.6/68.2	50.1/69.5
	APFL [Deng *et al.*, 2020]	29.5/38.7	46.0/67.7	38.6/64.2	55.7/75.7	56.2/67.1	59.7/68.2
	FedAMP [Huang *et al.*, 2021]	27.1/37.4	46.7/69.7	39.2/61.0	51.2/73.1	51.5/67.9	52.1/69.3
	FedATT [Jiang *et al.*, 2020]	30.5/40.8	58.7/79.7	38.4/63.7	52.4/79.1	50.9/70.0	53.5/72.6
	pFedMe [T Dinh *et al.*, 2020]	28.2/39.7	47.5/69.9	38.5/61.4	**50.5**/74.1	48.4/66.9	51.2/68.8
	SFL [Chen *et al.*, 2022]	31.1/39.2	46.4/68.8	37.6/59.3	54.2/73.7	47.2/66.0	49.8/67.2
	FedPoD (Ours)	**23.7/32.9**	**44.3/65.5**	**35.7/55.0**	51.4/**71.2**	**43.9/62.5**	**45.2/63.9**
Other Prompt Tuning	PromptFL [Guo *et al.*, 2023]	33.8/42.7	49.2/70.0	44.1/63.2	59.7/78.9	51.1/73.7	58.2/69.2
	MetePFL [Chen *et al.*, 2023b]	29.9/37.2	46.1/68.0	40.1/58.6	51.3/73.0	48.4/67.7	52.4/67.6

Table 2: Main results with different local fine-tuning strategy (MAE/RMSE reported), including Conventional Fine-tuning and ours adaptive prompt tuning, a lower value means better performance. **Bold** and Underline denote the best and second best respectively.

Variant	P_V	P_T	W_{bv}	W_{bt}	P_S	Federated Aggregation Strategy	Local Loss	**Task 1**	**Task 2**
FedPoD-A	*w/o*	*w*	-	-	*w*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{T}}\{P_i\}_{i=1}^{N} + (1-\alpha)\boldsymbol{A}_{\mathrm{S}}\{P_i\}_{i=1}^{N}$	Ours	29.9/40.4	53.7/78.4
							MSE	31.7/42.4	54.4/80.0
FedPoD-B	*w*	*w/o*	-	-	*w*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{S}}\{P_i\}_{i=1}^{N} + (1-\alpha)\boldsymbol{A}_{\mathrm{V}}\{P_i\}_{i=1}^{N}$	Ours	28.2/37.2	57.1/85.0
							MSE	29.2/39.0	58.2/85.9
FedPoD-C	*w/o*	*w/o*	-	-	*w*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{S}}\{P_i\}_{i=1}^{N}$	Ours	30.8/41.2	52.0/77.7
							MSE	31.8/42.4	54.8/78.9
FedPoD-D	*w*	*w*	*w*	*w*	*w/o*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{TV}}\{P_i\}_{i=1}^{N}$	Ours	30.1/40.9	48.7/74.7
							MSE	31.6/42.1	50.9/76.0
FedPoD-D	*w*	*w/o*	-	-	*w/o*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{V}}\{P_i\}_{i=1}^{N}$	Ours	29.4/39.8	56.2/84.7
							MSE	31.1/40.8	59.0/87.8
FedPoD-E	*w/o*	*w*	-	-	*w/o*	$\{P_i\}_{i=1}^{l,N} \leftarrow \boldsymbol{A}_{\mathrm{T}}\{P_i\}_{i=1}^{N}$	Ours	30.1/40.6	53.7/79.0
							MSE	31.7/43.5	54.2/80.5
FedPoD (Ori.)	*w*	*w*	*w*	*w*	*w*	$\{P_i\}_{i=1}^{l,N} \leftarrow \alpha\boldsymbol{A}\{P_i\}_{i=1}^{N} + (1-\alpha)\boldsymbol{A}'\{P_i\}_{i=1}^{N}$	Ours	**23.7/32.9**	**44.3/65.5**
							MSE	**25.0/34.4**	**47.7/68.0**

Table 3: Ablation results (MAE/RMSE report) about (1) Local Adaptive Prompts and (2) Local Optimization Objective, a lower value means better performance. **Bold**: the best, Underline: the second best, *w* and *wo* denote the presence and absence of prompt, respectively. Note that $\boldsymbol{A}_{\mathrm{T}}$ and $\boldsymbol{A}_{\mathrm{V}}$ are generated by Eq. 8 when either $\boldsymbol{P}_T$ or $\boldsymbol{P}_V$ is present alone.

6.1 Main Results

Table 2 presents main results, showing that **FedPoD** outperforms baselines in most scenarios, often by a significant margin, across various tuning strategies. Notably, our *adaptive prompt tuning* outperforms conventional fine-tuning while using about **74%** parameters (see Table 1). This method enhances baseline models by enabling them to learn fewer parameters for considerable performance boosts. **FedPoD** records an average performance increase of **23.6%/12.9%**, **11.7%/19.7%**, and **12.6%/4.3%** over FedAvg, FedProx, and Per-FedAvg, respectively. These percentages reflect MAE improvements for Task1/Task2. The gains are particularly striking against SFL, which employs graph-based aggregation [Chen *et al.*, 2022]. With adaptive prompt tuning, **FedPoD** improves by **9.8%** and **6.7%** on average. These figures rise to **15.9%** for Task1 and **11.1%** or Task2 with conventional fine-tuning. In addition, **FedPoD** can show a superior performance relative to FL-based prompt methods, PromptFL [Guo *et al.*, 2023] and MetePFL [Chen *et al.*, 2023b]. We credit these benefits to two main strategies: (1) *adaptive prompt tuning* guides the PFM to generate more accurate prediction based on lightweight prompts with multi-level communication, and (2) *dynamic graph modeling* encourages collaborative optimization among clients with similarly distribution to mitigate data heterogeneity. These components effectively address data heterogeneity and homogeneity issues through a lightweight plug-and-play means.

6.2 Framework Analysis

Ablation Study. We present the ablation results from two angles: (1) examining local prompts and their aggregation method, and (2) assessing the local optimization objective. This helps confirm the effectiveness of our Adaptive Prompt Tuning and Dynamic Graph Modeling. For (1), Table 3 reveal that: (i) ours local optimization objective outperforms MSE in all ablation scenarios concerning prompts, and (ii) the lack of any kind of prompt significantly hinders performance due to inadequate local representation and global dynamic aggregation. Furthermore, the impact of our multi-task optimization objective is detailed in Table 4, with Term 1: $\frac{1}{\xi^2}L^2(\{\boldsymbol{P}_i\},\{\boldsymbol{P}\}^*)$, Term 2: $\frac{1}{\xi^2}L^2(\{\boldsymbol{P}_i\},\{\boldsymbol{P}_i\}^l)$, Term 3: $\frac{1}{\tau^2}\cdot\frac{1}{(|\mathcal{N}|/\mathcal{S}_G)-1}\sum_{j\in\mathcal{N}}L^2(\{\boldsymbol{P}_i\},\{\boldsymbol{P}_j\}^l)$. This indicates that omitting any single term of our local optimization objective leads to a drop in overall performance, underscoring the importance and necessity of each component.

Term 1	Term 2	Term 3	Task 1	Task 2
w	*wo*	*wo*	29.1/36.9	47.1/70.1
w	*wo*	*w*	**27.3/36.3**	**46.0/69.9**
w	*w*	*wo*	29.1/34.3	<u>46.6/72.5</u>
wo	*w*	*w*	29.0/34.6	47.9/74.8
wo	*wo*	*w*	<u>28.2/37.0</u>	49.2/74.4

Table 4: Ablation results about the multi-task optimization objective (MAE/RMSE report). **Bold**: the best, <u>Underline</u>: the second best.

Privacy. We've implemented differential privacy (DP) in **FedPoD** by adding random noise to the gradient updates. This noise is scaled by a factor of $\tau = 1e^{-2}$. Table 5 shows a drop in performance after incorporating DP. Despite this, as shown in Table 2, **FedPoD** continues to surpass other baselines. Importantly, since **FedPoD** only uses adaptive prompts on the server to create graphs that capture the spatial-temporal relationships among clients, applying DP exclusively to these prompts is enough to maintain privacy.

Method/Dataset		**FedPoD**	**FedPoD-DP**	Ave. Variation
AvePRE	Task1	23.7/32.9	24.8/33.9	↓ 4.33%
	Task2	44.3/65.5	46.1/66.9	↓ 2.88%
SurTEMP	Task1	35.7/55.0	37.0/56.6	↓ 2.69%
	Task2	51.4/71.2	52.7/73.0	↓ 2.53%
SurUPS	Task1	43.9/62.5	45.1/63.7	↓ 2.33%
	Task2	45.2/63.0	46.4/65.2	↓ 2.34%

Table 5: Differential privacy experiment results (MAE/RMSE report), FedPoD-DP denotes FedPoD with DP.

Hyper-parameter Sensitivity. We examine the impact of hyper-parameters from two angles: the *prompt updating step* and the *subgraph step*. Our configuration is as follows: 5 local epochs and 10 communication rounds, while other settings follow main experiments. Table 6 shows that the best performance for Task 2 is achieved with a step of 1, and for Task 1 with a step of 6. This inconsistency is due to the variable nature of weather patterns. Additionally, setting the step to 12 results in the poorest performance for both Task 1 and Task 2. This is because a single-step update does not account for the erratic periodicity of weather patterns, leading to inflexibility. Our findings on the impact of $\mathcal{S}_G$ are shown in Table 7, where $\mathcal{S}_G \in \{1, 2, 4, 6, 8, 10\}$. The results suggests that **FedPoD** achieve the suboptimal when $\mathcal{S}_G = 1$ across different tasks, while optimal results are achieved for Task 1/2 when $\mathcal{S}_G = 10/2$. Bigger $\mathcal{S}_G$ means more knowledge will be involved in local optimization. In our experiment, not all clients train in each round for training due to the considerable overhead. With a large $\mathcal{S}_G$, clients are optimized locally with a restricted range of client knowledge, potentially overlooking valuable input from other participants and negatively affecting performance. We set $\mathcal{S}_G = 1$ as the default because it considers all clients and allows for flexibility in specific scenarios. As only prompts $\boldsymbol{P}_T, \boldsymbol{P}_S, \boldsymbol{P}_V$, which have fewer parameters, are involved, $\mathcal{S}_G = 1$ does not significantly increase communication costs.

Updating step of $\boldsymbol{P}_T$	Updating step of $\boldsymbol{P}_V$	MAE	RMSE
1	1	39.9/**51.5**	50.2/**79.5**
2	2	38.1/53.7	48.8/85.0
3	3	<u>37.1</u>/52.9	47.9/80.4
4	4	38.6/53.2	<u>47.7</u>/<u>80.1</u>
6	6	**35.7**/<u>52.6</u>	**46.1**/80.7
12	12	39.3/53.7	50.3/84.8

Table 6: Impact of prompt updating steps, **Bold**: the best, <u>Underline</u>: the second best, a lower value means a better performance. Note that the value format like "38.1/53.7" in the report denotes results for Task 1 and Task 2, respectively.

Step of subgraph $\mathcal{S}_G$	MAE (Task 1/Task 2)	RMSE (Task 1/Task 2)
1	<u>36.9</u>/<u>51.7</u>	<u>47.0</u>/79.4
2	39.1/**51.5**	49.9/**79.0**
4	38.1/51.8	49.7/<u>78.8</u>
6	38.8/54.0	49.1/81.9
8	39.3/54.4	49.7/81.8
10	**35.9**/52.4	**45.6**/79.6

Table 7: Results about impact of subgraph step $\mathcal{S}_G$, **Bold**: the best, <u>Underline</u>: the second best. Lower means bette performance.

7 Conclusion and Future Works

In this paper, we seek to tackle data heterogeneity among devices and data homogeneity within individual devices in on-device weather forecasting. To achieve this, we propose **FedPoD**, which is built on Adaptive Prompt Tuning and Dynamic Graph Modeling. The former aims to mitigate data homogeneity via extracting latent knowledge with the frozen foundation model, alongside multi-level communication, and the last deals with data heterogeneity by prioritizing devices with similar distribution for aggregation and collaborative training based on prompt-related graphs. Extensive experiments on real-world on-device weather forecasting datasets shows **FedPoD** consistently outperforms state-of-the-art methods. However, **FedPoD** can struggle with long-term predictions due to the gradual updating of prompts. We plan to address this limitation in future research and extend our approach to more on-device spatiotemporal reasoning challenges, such as forecasting and imputation.

References

[Chakraborty and Rodrigues, 2020] Chinmay Chakraborty and Joel JCP Rodrigues. A comprehensive review on device-to-device communication paradigm: trends, challenges and applications. *Wireless Personal Communications*, 114(1):185–207, 2020.

[Chavan and Momin, 2017] Gaurav Chavan and Bashirahamad Momin. An integrated approach for weather forecasting over internet of things: A brief review. In *2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, pages 83–88. IEEE, 2017.

[Chen and Lai, 2011] Ling Chen and Xu Lai. Comparison between arima and ann models used in short-term wind speed forecasting. In *2011 Asia-Pacific Power and Energy Engineering Conference*, pages 1–4. IEEE, 2011.

[Chen *et al.*, 2022] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with graph. *arXiv preprint arXiv:2203.00829*, 2022.

[Chen *et al.*, 2023a] Shengchao Chen, Guodong Long, Jing Jiang, Dikai Liu, and Chengqi Zhang. Foundation models for weather and climate data understanding: A comprehensive survey. *arXiv preprint arXiv:2312.03014*, 2023.

[Chen *et al.*, 2023b] Shengchao Chen, Guodong Long, Tao Shen, and Jing Jiang. Prompt federated learning for weather forecasting: Toward foundation models on meteorological data. *arXiv preprint arXiv:2301.09152*, 2023.

[Chen *et al.*, 2023c] Shengchao Chen, Ting Shu, Huan Zhao, Guo Zhong, and Xunlai Chen. Tempee: Temporal-spatial parallel transformer for radar echo extrapolation beyond auto-regression. *arXiv preprint arXiv:2304.14131*, 2023.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

[Deng *et al.*, 2020] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[Fallah *et al.*, 2020] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[Grover *et al.*, 2015] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 379–386, 2015.

[Guo *et al.*, 2023] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 2023.

[Hagemann *et al.*, 2013] Stefan Hagemann, Cui Chen, Douglas B Clark, Sonja Folwell, Simon N Gosling, Ingjerd Haddeland, Naota Hanasaki, Jens Heinke, Fulco Ludwig, Frank Voss, et al. Climate change impact on available water resources obtained using multiple global climate and hydrology models. *Earth System Dynamics*, 4(1):129–144, 2013.

[Hanzely *et al.*, 2020] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020.

[Huang *et al.*, 2021] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7865–7873, 2021.

[Jiang *et al.*, 2020] Jing Jiang, Shaoxiong Ji, and Guodong Long. Decentralized knowledge acquisition for mobile internet applications. *World Wide Web*, 23(5):2653–2669, 2020.

[Karl *et al.*, 2009] Thomas R Karl, Jerry M Melillo, and Thomas C Peterson. *Global climate change impacts in the United States: a state of knowledge report from the US Global Change Research Program*. Cambridge University Press, 2009.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[Kjellstrom *et al.*, 2016] Tord Kjellstrom, David Briggs, Chris Freyberg, Bruno Lemke, Matthias Otto, and Olivia Hyatt. Heat, human performance, and occupational health: a key issue for the assessment of global climate change impacts. *Annual review of public health*, 37:97–112, 2016.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[Li *et al.*, 2021a] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[Li *et al.*, 2021b] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Paulik *et al.*, 2021] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*, 2021.

[Robusto, 1957] C Carl Robusto. The cosine-haversine formula. *The American Mathematical Monthly*, 64(1):38–40, 1957.

[Sapankevych and Sankar, 2009] Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE computational intelligence magazine*, 4(2):24–38, 2009.

[Shi *et al.*, 2015] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[T Dinh *et al.*, 2020] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

[Tan *et al.*, 2022] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pretrained models: A contrastive learning approach. *arXiv preprint arXiv:2209.10083*, 2022.

[Voyant *et al.*, 2012] Cyril Voyant, Marc Muselli, Christophe Paoli, and Marie-Laure Nivet. Numerical weather prediction (nwp) and hybrid arma/ann model to predict global radiation. *Energy*, 39(1):341–355, 2012.

[White *et al.*, 2023] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[Xiong *et al.*, 2023] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16323–16332, 2023.

[Yu *et al.*, 2017] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[Zhang *et al.*, 2020] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*, 2022.