Plag-Llama for IoT-Enabled Data Centers: A Multivariate Time Series Forecasting Approach

Yu Sun*, Bo Cheng*, Jinan Li†, Gaoxiang Jiang*, Tianqi Zhang*, and Haibo Zhou*
*School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210023.
Emails: {yusun,bocheng,gaoxiangjiang,tianqizhang}@smail.nju.edu.cn, haibozhou@nju.edu.cn.
† Guangdong Branch, China Unicom Group Co., Ltd., Guangzhou, China, 510627.
Email: eljn@foxmail.com

GLOBECOM 2024 - 2024 IEEE Global Communications Conference | 979-8-3503-5125-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GLOBECOM52923.2024.10901348

***Abstract*—By harnessing advanced Internet-of-Things (IoT) technologies, deep-learning methods have garnered significant attention for accurately predicting data center status, serving as the cornerstone for mitigating the exponential growth of energy consumption in data centers. However, these methods encounter data scarcity issues during practical deployment. While the proliferation of large models holds promise for addressing this challenge, their application in the context of data centers remains largely unexplored. Furthermore, these models encounter diverse obstacles, including multivariate tasks, immersive computation, and so on. In this paper, we investigate multivariate time series forecasting in IoT-enabled data centers by harnessing large models. Specifically, we introduce a multivariate time series forecasting framework tailored for IoT-enabled data centers. We propose the point Lag (Plag)-Llama model, which transfers univariate forecasting knowledge from the Lag-Llama for multivariate forecasting. The Plag-Llama benefits from zero-shot ability and fine-tuning facilitated by our proposed transfer block. To mitigate computational intensity and enhance the Plag-Llama's performance, we introduce a novel Joint Channel-Time (JCT)-adapter fine-tuning technique. Extensive experiments demonstrate that the transferred Plag-Llama exhibits superior zero-shot ability, while the proposed JCT-adapter fine-tuning achieves state-of-the-art performance and remarkable few-shot ability on real-world datasets collected from data centers.**

***Index Terms*—Multivariate time series, large models, adapter fine-tuning, Internet-of-Things, data center.**

I. Introduction

Data centers hold a crucial position in global energy consumption and carbon emissions, with their importance projected to escalate exponentially, especially with the widespread adoption of large-scale models like ChatGPT and Llama [1]. Concurrently, governments and data center operators vigorously enact policies and initiatives targeted at curbing energy consumption [2]. The optimization of cooling systems has garnered significant attention, given their substantial share of energy consumption in data centers. Precise modeling of data centers plays a pivotal role in this context, serving as the key foundation for subsequent optimization algorithms [3]. Recently, the evolution of Internet-of-Things (IoT) techniques [4] has facilitated data-driven modeling, showcasing superiority in providing accurate descriptions of future data center status compared to traditional methods. Vu et al. [5] employed distinct deep-learning algorithms tailored to specific module types within a module-wise modeling approach, predicting power, water flow, and water temperature. Similarly, Jiang et al. [6] adopted multi-task learning for multiple prediction tasks. In contrast, Zhao et al. [7] relied on a gated recurrent unit-based approach, leveraging historical data to predict power usage effectiveness. However, a critical aspect remains unaddressed in these works: multivariate time series forecasting. Despite their successes, these methods encounter challenges related to data scarcity during practical deployment, requiring sufficient data accumulation after all IoT devices are operational, thereby incurring significant time costs.

With the success of large (or foundation) models across various domains, recent studies have also explored their significant capabilities in time series forecasting. These models demonstrate superior zero-shot and few-shot abilities, which are anticipated to mitigate the data scarcity challenge, thereby facilitating swift deployment in data centers. Several methodologies have been investigated for Large Language Models (LLMs), including prompt-based approaches [8], reprogramming techniques [9], and two-stage fine-tuning strategies [10], with the objective of aligning time and text modalities and achieving robust performance in time series forecasting. Furthermore, foundation models specifically tailored for time series forecasting have been developed. TimeGPT [11], trained on 100 billion data points, exhibits accurate prediction and strong zero-shot capabilities. Lag-Llama [12], targeting probabilistic univariate time series forecasting, has demonstrated superior performance across diverse time series datasets. However, these methods encounter various challenges to different extents. For example, prompt-based LLMs may struggle with multivariate time series, and the high cost of LLMs presents a barrier to widespread adoption. Furthermore, TimeGPT, exclusively available via API, is not suitable for deployment in production environments. Additionally, there has been no prior investigation into the realm of data center applications. In particular, Lag-Llama, a cutting-edge time series foundation model designed for univariate probabilistic forecasting, encounters several challenges when applied to multivariate time series in data centers. These challenges include: 1) transferring from probabilistic to point forecasting, 2) empowering the multivariate forecasting capabilities of Lag-Llama, and 3) addressing computational intensity and overfitting issues encountered during fine-tuning.

In this paper, we investigate multivariate time series forecasting in IoT-enabled data centers to overcome the challenge of data scarcity and achieve state-of-the-art performance. We begin by defining the multivariate time series forecasting problem specific to data centers. Then, we introduce the point Lag (Plag)-Llama model, transferred from Lag-Llama, which facilitates zero-shot ability and fine-tuning for multivariate point time series forecasting. Furthermore, we propose the joint channel-time adapter fine-tuning technique to alleviate computational intensity and enhance model performance. Finally, we evaluate these methods against state-of-the-art approaches through extensive experiments on real-world datasets. The main contributions are highlighted as follows:

- We construct a multivariate time forecasting framework for IoT-enabled data centers, employing the large model to address data scarcity issues, facilitating rapid deployment, and achieving superior performance. To our knowledge, this represents the inaugural endeavor to employ large models within the domain of data centers.
- We transfer the capabilities of the Lag-Llama model from probabilistic univariate forecasting to multivariate point time forecasting, enabling zero-shot ability and full fine-tuning for new tasks.
- We introduce a novel Joint Channel-Time (JCT)-adapter fine-tuning method to proficiently adapt the model to multivariate time series forecasting, addressing computational intensity and overfitting challenges while achieving state-of-the-art performance.

The rest of this paper is organized as follows. Section II provides an introduction to multivariate time series forecasting in IoT-enabled data centers. The Plag-Llama architecture and JCT-adapter fine-tuning are elaborated upon in Section III. The experimental results are presented in Section IV, followed by the conclusion in Section V.

II. Problem Definition

A. Multivariate Time Series Forecasting in IoT-Enabled Data Centers

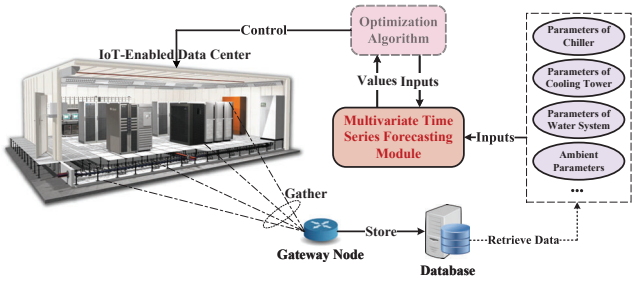

Fig. 1: A framework for multivariate time series forecasting in IoT-enabled data centers.

The framework for multivariate time series forecasting in IoT-enabled data centers is illustrated in Fig. 1. Leveraging advanced IoT technologies, we can monitor the real-time status of data centers by deploying diverse sensors, including those for temperature, humidity, and flow measurement. After deploying all IoT devices, continuous data streams are sensed, collected, and processed by gateway nodes before storage in the database. Over time, as sufficient data accumulates, we retrieve this information—including parameters related to chillers, cooling towers, water systems, ambient conditions, and more—as training data for constructing the multivariate time series forecasting module. Notably, one of our primary motivations is to leverage the few-shot and zero-shot capabilities of foundation models, thus shortening the data accumulation period. Precisely predicting the status of data centers lays the foundation for optimization algorithms[1]. We frame this task as a multivariate time series forecasting problem, capitalizing on the time-dependent intrinsic characteristics and interdependencies among variables.

Let $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots\}$ denote the multivariate time series, where $\boldsymbol{x}_t \in \mathbb{R}^C$ represents the multivariate vector with dimension C (i.e., number of variates or channels) at time t, expressed as $\boldsymbol{x}_t = [x_1, x_2, \ldots, x_C]^T$. The multivariate time series forecasting problem in data centers is defined as follows: Given an L-length history window $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{C \times L}$, the task aims to accurately predict the future H time steps, i.e., $\mathbf{X}_{t+1:t+H} \in \mathbb{R}^{C \times H}$, by minimizing the following objective:

$$\mathcal{L}\left(\mathbf{X}_{t+1:t+H}, \hat{\mathbf{X}}_{t+1:t+H}\right), \tag{1}$$

where $\mathcal{L}$ represents the loss function, and $\hat{\mathbf{X}}_{t+1:t+H}$ is derived from the multivariate time series forecasting model $\mathcal{M}$:

$$\hat{\mathbf{X}}_{t+1:t+H} = \mathcal{M}\left(\mathbf{X}_{t-L+1:t}\right). \tag{2}$$

III. Model Architecture

A. Lag-Llama

Lag-Llama [12], a foundational model targeting probabilistic univariate time series forecasting, is built upon the decoder-only transformer-based LLaMA architecture, which integrates lagged features. It has demonstrated superior zero-shot ability and state-of-the-art performance across diverse time series datasets. However, it is crucial to recognize that Lag-Llama is designed specifically for univariate forecasting. The superior performance, as emphasized in [12], arises when multivariate time series are processed and forecasted as multiple independent univariate sequences across all benchmark models. Consequently, the adaptation of Lag-Llama to multivariate time series forecasting remains uncertain. Furthermore, in our data center scenarios, our primary requirement is accurate prediction of future status rather than the entire distribution. This necessitates certain modifications to Lag-Llama. Moreover, our experiments with the real-world dataset reveal that Lag-Llama suffers from overfitting. Surprisingly, fine-tuning exacerbates performance rather than improving it. Additionally, the full fine-tuning process for Lag-Llama is computationally intensive and resource-demanding.

B. Plag-Llama: Transferring Lag-Llama to Multivariate Time Series Point Forecasting

We introduce Plag-Llama, a model crafted for multivariate time series point forecasting, which extends the capabilities of

[1]The research of optimization algorithms is beyond the scope of this paper.

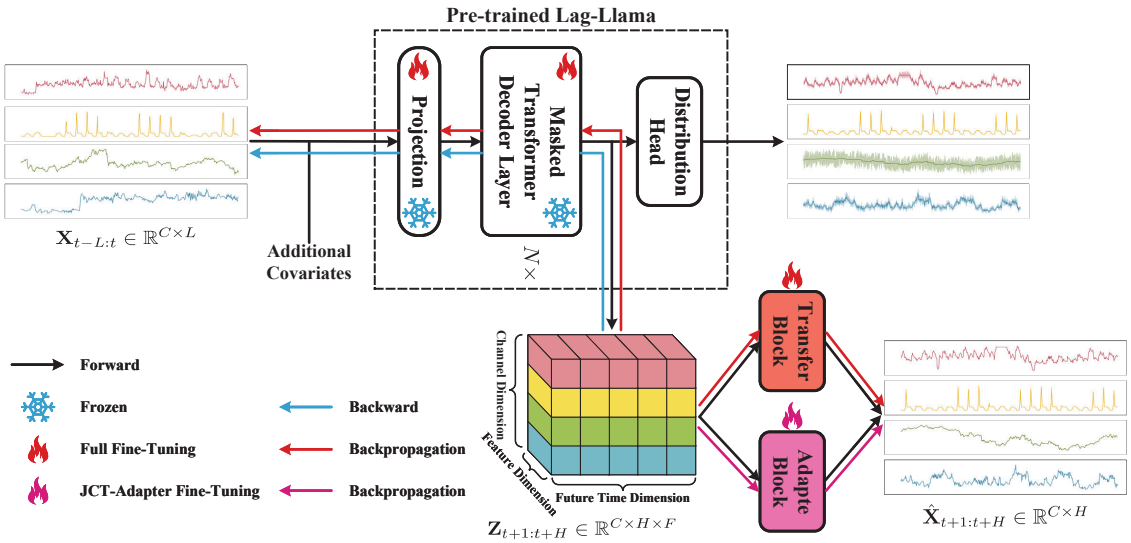

Fig. 2: Plag-Llama: Overview of the architecture.

the pre-trained Lag-Llama. The overall architecture is depicted in Fig. 2. In Lag-Llama, historical inputs $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{C\times L}$ undergo sequential processing involving a projection layer, N masked transformer decoder layers, and a distribution head, yielding a probabilistic distribution for each time series. To adapt the probabilistic forecasting capability to point forecasting, the pre-trained Lag-Llama, excluding the distribution head, serves as the backbone of Plag-Llama. Furthermore, Plag-Llama integrates supplementary blocks, such as the transfer block and adapter block, incorporates a revised loss function, and employs fine-tuning techniques.

C. Full Fine-Tuning

Fine-tuning, as an effective transfer mechanism, has garnered significant attention in the research domain. One of the fundamental methods, the full fine-tuning approach, is also employed in Lag-Llama, wherein all the parameters of the pre-trained model are tuned for new downstream tasks. However, it is important to note that the full fine-tuning strategy used here for Plag-Llama differs from that in Lag-Llama due to the distinct forecasting objectives. Specifically, in Plag-Llama, the distribution head is replaced by a transfer block. Furthermore, the negative log-likelihood loss, typically employed for probabilistic forecasting, is appropriately substituted with the Mean Squared Error (MSE) loss for point forecasting.

Transfer blocks can be implemented in various ways. In this work, we propose an intuitive and zero-shot block that leverages the average operation across feature dimensions:

$$\hat{\mathbf{X}}_{t+1:t+H} = \text{Average}\left(\mathbf{Z}_{t+1:t+H}\right), \tag{3}$$

where $\mathbf{Z}_{t+1:t+H}$ represents the output from the backbone model. The averaging operation is performed along the feature dimension, resulting in $\hat{\mathbf{X}}_{t+1:t+H}$, which directly corresponds to the prediction values. The introduction of the feature-averaging trick is primarily grounded in the consideration of two key factors. Firstly, the foundation model excels in its zero-shot capability, which can be reacquired through this straightforward method. Secondly, it is essential to acknowledge that a more intricate structure might negatively impact performance during full fine-tuning, given that such a simple trick is also susceptible to overfitting. In the context of Plag-Llama, the update rule for full fine-tuning can be succinctly described as follows:

$$(\boldsymbol{\omega}_{i+1}, \boldsymbol{\theta}_{i+1}) = (\boldsymbol{\omega}_i, \boldsymbol{\theta}_i) - \alpha \nabla \mathcal{L}\left(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i\right), \tag{4}$$

where i denotes the training step, α represents the learning rate, $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ correspond to the parameters of the backbone and transfer block, respectively, and $\mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\theta}\right)$ denotes the loss function. For a comprehensive understanding of the entire full fine-tuning process, encompassing forward, backpropagation, and module update status, please refer to Fig. 2.

D. Joint Channel-Time Adapter Fine-Tuning

In above full fine-tuning, all parameters $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ must be adjusted for each task, resulting in computational intensity and resource demands. Despite employing early stopping and random sampling strategies, full fine-tuning remains susceptible to overfitting in our multivariate time series forecasting experiments. Parameter-efficient fine-tuning offers a promising approach to address this weakness by tuning only specific parameters. However, existing works [13] struggle to capture new relationships beyond the original model. Specifically, they fail to account for new channel dependencies required for multivariate time series forecasting based on univariate models. Similarly, this capability is absent in the univariate Lag-Llama model itself. Unfortunately, this absence of information may significantly impact performance.

In order to address the aforementioned challenges, we propose a novel method, termed JCT-adapter fine-tuning, which aims to capture interdependencies across channels and time steps by integrating a new adapter block. This approach facilitates the aggregation of global channel-time

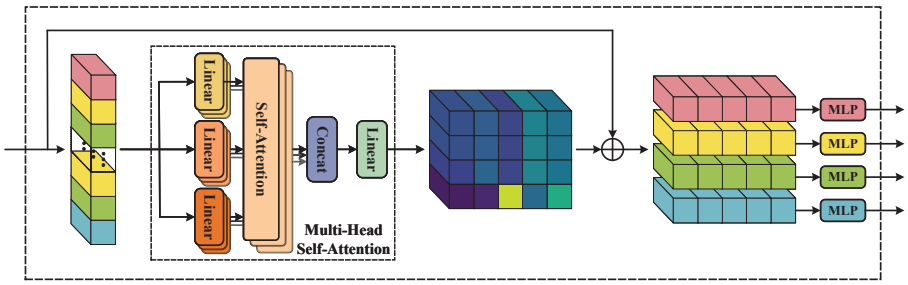

Fig. 3: Architecture of the joint channel-time (JCT)-adapter block.

information across diverse channels and at various time steps. Consideration of the placement of the inserted module is crucial, especially in light of the challenges presented by Lag-Llama. These challenges encompass the transfer from univariate to multivariate forecasting, from probabilistic to point forecasting, as well as computational intensity and overfitting. Fortunately, the adapter inherently addresses concerns regarding point forecasting and overfitting. However, multivariate forecasting demands the capture of inter-channel dependencies, necessitating substantial information and potentially resulting in higher computational overhead. Therefore, well-balanced trade-offs are essential for optimal performance. In light of these considerations, we strategically position the adapter module after the masked transformer decoder layer, as depicted in Fig. 2. This placement ensures that informative channel, time, and feature dimensions are preserved in the outputs. Notably, the time dimension exclusively represents future values, eliminating historical data and thereby reducing computational load.

The specific architecture of the JCT-adapter is shown in Fig. 3. Initially, the output from the backbone, denoted as $\mathbf{Z}_{t+1:t+H}$, is flattened to form $\mathbf{Y} \in \mathbb{R}^{CH \times F}$. Subsequently, this flattened representation passes through a multi-head self-attention block, where a residual connection is also incorporated. Finally, the predicted values are derived after passing through their respective MLP layers. The flattening operation is employed with the intention of adequately capturing the correlations between each time step across all channels, necessitating a computational cost of $\mathcal{O}\left(C^2H^2\right)$. This cost, although notable, is deemed acceptable as it only involves future time information, rather than including historical time data, which would incur a substantially higher cost of $\mathcal{O}\left(C^2(L+H)^2\right)$. The formal definition of multi-head self-attention is as follows:

$$\mathcal{T}_M = [\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_h]\,\mathbf{O}, \tag{5a}$$

$$\text{where } \mathcal{T}_i = \mathcal{T}\left(\mathbf{Y}\mathbf{W}_i^Q, \mathbf{Y}\mathbf{W}_i^K, \mathbf{Y}\mathbf{W}_i^V\right), \tag{5b}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V \in \mathbb{R}^{CT \times d_i}$ represent linear transformations for the query, key, and value in the i-th head, respectively. The self-attention mechanism $\mathcal{T}$ is computed as follows:

$$\mathcal{T} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}}\right)\mathbf{V}. \tag{6}$$

The concept of multi-head attention draws inspiration from the mixture of experts (MoE) framework employed in certain adapter designs [13]. In our context, each expert in the MoE corresponds to one head of multi-head attention, responsible for capturing specific correlations from distinct aspects. Consequently, the output projection in Eq. (5a) aggregates information contributed by multiple experts.

During the fine-tuning process of the JCT-adapter for new tasks, we selectively update parameters. Specifically, only the newly introduced parameters $\boldsymbol{\nu}$ within the JCT-adapter block are adjusted, while the original parameters $\boldsymbol{\omega}$ remain frozen. This adjustment entails a revision of the update rule to:

$$\boldsymbol{\nu}_{i+1} = \boldsymbol{\nu}_i - \alpha \nabla \mathcal{L}'\left(\boldsymbol{\nu}_i\right), \tag{7}$$

where $\mathcal{L}'\left(\boldsymbol{\nu}\right)$ represents the Huber loss, selected for its robustness compared to MSE. The Huber loss is defined as:

$$\text{HuberLoss}\left(y, \hat{y}\right) = \begin{cases} \frac{1}{2}\left(y - \hat{y}\right)^2, & \text{if } |y - \hat{y}| \le \delta, \\ \delta\left(|y - \hat{y}| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases} \tag{8}$$

The fine-tuning procedure for the JCT-adapter is depicted in Fig. 2. In contrast to full backpropagation, which updates all parameters, our proposed approach selectively tunes the adapter block while maintaining the backbone frozen. As a result, backpropagation terminates at the adapter block, and no gradients are propagated to the shallower layers.

IV. Experiments

A. Experimental Settings

Dataset. We utilize data collected from a real IoT-enabled data center [2], employing similar data processing methods. The processed multivariate time series consist of one-minute intervals and 130 dimensions, totaling 274,662 samples.

Baselines. We establish the following recent state-of-the-art methods as baselines: Autoformer [14], Crossformer [15], DLinear [16], FEDformer [17], Informer [18], iTransformer [19], LightTS [20], Non-stationary (NS)-Transformer [21], PatchTST [22], TimesNet [23], and Transformer [2]. Ablation studies are conducted on JCT-adapter fine-tuning by removing the attention block, denoted as adapter. The history length is set to $L = 8$, the prediction length to $H = 4$, with a batch size of 512 and a learning rate of 3e-4. We utilize 8 attention heads, and the hidden layer sizes of the MLP are set to 128, 64, and 32, respectively. Adam optimizer is employed, while

the other parameters adhere to their originally recommended values.

Metrics. We employ commonly used metrics in time series forecasting, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), and Mean Absolute Scaled Error (MASE). Additionally, we calculate the average rank (ARk) across these metrics.

B. Experimental Results

TABLE I: Forecasting Performance Comparisons (Best results in **bold**, second best underlined, third best dotted underline).

MODEL	MAE	RMSE	SMAPE	MASE	ARk
SUPERVISED					
Autoformer	0.045	0.188	6.729	0.039	10.25
Crossformer	0.078	0.244	9.264	0.068	12.00
DLinear	0.041	0.196	5.664	0.036	6.50
FEDformer	0.044	0.167	6.646	0.038	8.25
Informer	0.082	0.625	9.392	0.071	13.25
iTransformer	0.041	0.184	6.020	0.035	5.50
LightTS	0.042	0.178	6.305	0.036	6.75
NS-Transformer	0.042	0.193	6.306	0.036	8.25
PatchTST	0.038	0.185	5.468	0.033	4.00
TimesNet	0.041	0.191	6.151	0.036	6.75
Transformer	0.670	1.045	72.235	0.580	15.00
ZERO-SHOT					
Plag-Llama	0.039	0.173	5.946	0.034	4.25
FINE-TUNING					
Plag-Llama (Full)	0.175	0.251	15.879	0.152	13.75
Plag-Llama (JCT-Adapter)	**0.035**	**0.155**	**5.280**	**0.030**	**1.00**
Plag-Llama (Adapter)	0.036	0.160	5.339	0.031	2.00

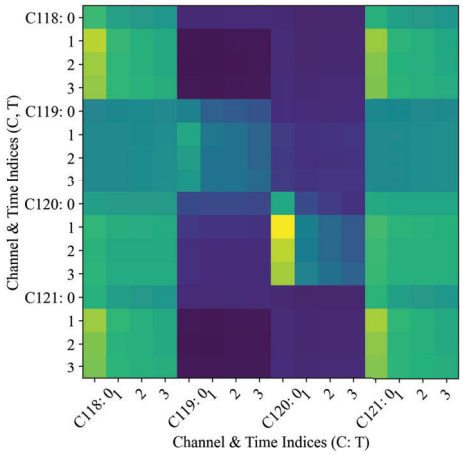

Fig. 4: Visualization of attention across channels and time.

*1) **Forecasting Performance**:* We provide a comparative analysis of forecasting performance between our proposed methods and state-of-the-art baselines, summarized in Table I. Empowered by the transfer block, Plag-Llama exhibits robust zero-shot ability, achieving an average rank of 4.25. Notably, it secures the second position among supervised methods. This remarkable performance underscores the feasibility of Plag-Llama, positioning it as a promising solution to address data scarcity challenges and expedite deployment within data centers. It is noteworthy that the prediction result of probabilistic forecasting is a distribution, representing a range of possible values along with associated probabilities. The goal of probabilistic forecasting is to bring the overall distribution closer to the ground truth, rather than focusing solely on aligning the mean values with the ground truth. However, the average pooling layer can effectively transfer the ability from probabilistic forecasting to point forecasting. This capability may be attributed to the symmetric Student's t-distribution adopted in Lag-Llama, where learning an accurate prediction of the ground truth as the mean parameters of the distribution could be a favorable choice.

Instead, employing the full fine-tuning method for Plag-Llama significantly undermines performance due to overfitting. Conversely, the proposed JCT-adapter fine-tuning consistently outperforms state-of-the-art methods across various metrics, exhibiting an average improvement of 9.16% over the best supervised methods. Additionally, the ablation study on multi-head self-attention reveals that removing the attention module within the JCT-adapter results in performance degradation of up to 3.33%. To further validate the effectiveness of the attention block in capturing interdependencies across channels and time, we visualize its attention weights within the JCT-adapter for channels 118 to 121 and their corresponding four time steps. As depicted in Fig. 4, significant correlations between different time steps and channels are observed. Of particular note is that the attention maps are segmented by channels, indicating similar correlations across different time steps for arbitrary channel pairs. While correlations within a channel over time steps do exist, their weights are smaller compared to those across channels. The results above demonstrate the effectiveness of the proposed JCT-adapter fine-tuning in capturing both channel and time dependencies in multivariate time series forecasting, thereby outperforming state-of-the-art methods. Additionally, the ablation study confirms the necessity of the attention module.

*2) **Few-Shot Ability**:* To assess the few-shot ability of the proposed methods, we specifically choose the final 20%, 40%, 60%, and 80% of the data as the training dataset for all benchmarks. To ensure readability and conserve space, we select four representative methods known for their strong performance: iTransformer and TimesNet for channel dependency, and DLinear and PatchTST for channel independence. The results are summarized in Table II. Across varying lengths of historical training data, JCT-adapter fine-tuning consistently demonstrates the best average performance, only ranking second on two metrics when data is limited. As the historical data increases, more information and structure can be captured, leading to progressively better performance in JCT-adapter fine-tuning. This trend aligns with expectations, with improvements compared to state-of-the-art methods increasing from 4.08% to 7.36%. However, methods like iTransformer and DLinear exhibit deteriorating performance with increasing data, highlighting their deficiencies. Overall, these results underscore the robust few-shot capability of the JCT-adapter, whose superiority becomes increasingly evident as more data is accumulated.

V. Conclusion

In this paper, we have investigated the multivariate time series forecasting problem in IoT-enabled data centers by

TABLE II: Few-Shot Ability Comparisons (Best results in **bold**, second best underlined, third best dotted underline).

DATA %	MODEL	MAE	RMSE	SMAPE	MASE	ARk	DATA %	MODEL	MAE	RMSE	SMAPE	MASE	ARk
20 %	DLinear	0.041	0.196	5.672	0.036	5.5	40 %	DLinear	0.041	0.196	5.696	0.036	5.0
	iTransformer	0.038	0.181	5.646	0.033	3.0		iTransformer	0.038	0.184	5.557	0.033	3.25
	PatchTST	0.039	0.188	5.645	0.034	4.0		PatchTST	0.039	0.187	5.491	0.034	3.75
	TimesNet	0.042	0.194	6.334	0.037	6.0		TimesNet	0.043	0.209	6.272	0.037	6.25
	Plag-Llama (Full)	0.054	**0.146**	7.385	0.047	5.5		Plag-Llama (Full)	0.137	0.206	13.223	0.119	6.75
	Plag-Llama (JCT-Adapter)	**0.037**	0.162	5.638	**0.032**	**1.5**		Plag-Llama (JCT-Adapter)	**0.036**	**0.159**	**5.454**	**0.031**	**1.0**
	Plag-Llama (Adapter)	0.038	0.162	**5.607**	0.033	2.5		Plag-Llama (Adapter)	0.037	0.162	5.477	0.032	2.0
DATA %	**MODEL**	**MAE**	**RMSE**	**SMAPE**	**MASE**	**ARk**	**DATA %**	**MODEL**	**MAE**	**RMSE**	**SMAPE**	**MASE**	**ARk**
60 %	DLinear	0.041	0.196	5.680	0.036	5.0	80 %	DLinear	0.041	0.196	5.674	0.036	5.25
	iTransformer	0.039	0.187	5.550	0.033	4.0		iTransformer	0.038	0.182	5.399	0.032	3.25
	PatchTST	0.038	0.185	5.514	0.033	3.0		PatchTST	0.038	0.186	5.472	0.033	4.25
	TimesNet	0.043	0.206	6.240	0.037	6.0		TimesNet	0.043	0.201	6.314	0.037	6.25
	Plag-Llama (Full)	0.635	0.915	61.296	0.549	7.0		Plag-Llama (Full)	0.069	0.164	8.770	0.059	6.0
	Plag-Llama (JCT-Adapter)	**0.036**	**0.156**	**5.371**	**0.031**	**1.0**		Plag-Llama (JCT-Adapter)	**0.035**	**0.156**	**5.344**	**0.030**	**1.0**
	Plag-Llama (Adapter)	0.036	0.162	5.422	0.032	2.0		Plag-Llama (Adapter)	0.036	0.161	5.386	0.031	2.0

leveraging the time series foundation model Lag-Llama. We have transferred Lag-Llama for the multivariate time series forecasting task, naming it Plag-Llama, where the zero-shot ability and fine-tuning are facilitated by the transfer block. We have proposed the Joint Channel-Time (JCT) adapter fine-tuning, addressing the challenges of data scarcity and those within Plag-Llama. Extensive experiments have demonstrated the superior zero-shot ability of Plag-Llama, as well as the state-of-the-art performance and remarkable few-shot ability of JCT-adapter fine-tuning. In summary, the proposed methods effectively address data scarcity challenges, enabling rapid deployment while achieving superior performance. In the future, we will conduct more experiments on public datasets to further validate the superiority and generalization ability of the proposed methods.

REFERENCES

[1] P. Li, H. Zhang, Y. Wu, L. Qian, R. Yu, D. Niyato, and X. Shen, "Filling the missing: Exploring generative AI for enhanced federated learning over heterogeneous mobile edge devices," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2024.

[2] Y. Sun, Y. Wang, G. Jiang, B. Cheng, and H. Zhou, "Deep learning-based power usage effectiveness optimization for IoT-enabled data center," *Peer-to-Peer Networking and Applications*, Mar. 2024.

[3] Y. Wang, Y. Sun, B. Cheng, G. Jiang, and H. Zhou, "DQN-based chiller energy consumption optimization in IoT-enabled data center," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, Oct. 2023, pp. 985–990.

[4] L. P. Qian, S. Zhou, M. Wu, and Y. Wu, "Joint optimization of resource allocation and SIC ordering in energy-harvesting relay-aided NOMA NB-IoT networks," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 1, pp. 468–481, Mar. 2024.

[5] H. D. Vu, K. S. Chai, B. Keating, N. Tursynbek, B. Xu, K. Yang, X. Yang, and Z. Zhang, "Data Driven Chiller Plant Energy Optimization with Domain Knowledge," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1309–1317.

[6] G. Jiang, Y. Sun, B. Cheng, Y. Wang, and H. Zhou, "Leveraging Multi-Task Learning for Energy Consumption Prediction in IoT-Based Data Center," in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, Oct. 2023, pp. 943–948.

[7] P. Zhao, L. Yang, Z. Kang, and J. Lin, "On Predicting the PUE with Gated Recurrent Unit in Data Centers," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, Dec. 2019, pp. 1664–1670.

[8] H. Xue and F. D. Salim, "PromptCast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.

[9] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, Oct. 2023.

[10] C. Chang, W.-C. Peng, and T.-F. Chen, "LLM4TS: Two-stage fine-tuning for time-series forecasting with pre-trained LLMs," *arXiv preprint arXiv:2308.08469*, Oct. 2023.

[11] A. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," *arXiv preprint arXiv:2310.03589*, Oct. 2023.

[12] K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish, "Lag-llama: Towards foundation models for probabilistic time series forecasting," *arXiv preprint arXiv:2310.08278*, Feb. 2024.

[13] V. Lialin, V. Deshpande, and A. Rumshisky, "Scaling down to scale up: A guide to parameter-efficient fine-tuning," *arXiv preprint arXiv:2303.15647*, Mar. 2023.

[14] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 22 419–22 430.

[15] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, Sep. 2022.

[16] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, pp. 11 121–11 128, Jun. 2023.

[17] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, Jun. 2022, pp. 27 268–27 286.

[18] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11 106–11 115, May 2021.

[19] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "Itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[20] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "LightTS: Lightweight time series classification with adaptive ensemble distillation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 171:1–171:27, Jun. 2023.

[21] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, Dec. 2022.

[22] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, Sep. 2022.

[23] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, Sep. 2022.