

LightLLM: A Versatile Large Language Model for Predictive Light Sensing

Jiawei Hu

University of New South Wales
CSIRO
Australia
jiawei.hu@unsw.edu.au

Lina Yao

CSIRO
Australia
lina.yao@data61.csiro.au

Hong Jia

University of Melbourne
Australia
hong.jia@unimelb.edu.au

Brano Kusy

CSIRO
Australia
brano.kusy@data61.csiro.au

Mahbub Hassan

University of New South Wales
Australia
mahbub.hassan@unsw.edu.au

Wen Hu

University of New South Wales
Australia
wen.hu@unsw.edu.au

ABSTRACT

We propose LightLLM, a model that fine tunes pre-trained large language models (LLMs) for light-based sensing tasks. It integrates a sensor data encoder to extract key features, a contextual prompt to provide environmental information, and a fusion layer to combine these inputs into a unified representation. This combined input is then processed by the pre-trained LLM, which remains frozen while being fine-tuned through the addition of lightweight, trainable components, allowing the model to adapt to new tasks without altering its original parameters. This approach enables flexible adaptation of LLM to specialized light sensing tasks with minimal computational overhead and retraining effort. We have implemented LightLLM for three light sensing tasks: light-based localization, outdoor solar forecasting, and indoor solar estimation. Using real-world experimental datasets, we demonstrate that LightLLM significantly outperforms state-of-the-art methods, achieving 4.4x improvement in localization accuracy and 3.4x improvement in indoor solar estimation when tested in previously unseen environments. We further demonstrate that LightLLM outperforms ChatGPT-4 with direct prompting, highlighting the advantages of LightLLM’s specialized architecture for sensor data fusion with textual prompts.

1 INTRODUCTION

Predictive light sensing (PLS) is becoming increasingly significant as the demand for intelligent and sustainable systems grows. For example, by leveraging light sensor signatures, PLS can accurately infer the location of individuals in indoor spaces, enabling seamless **indoor positioning** using light spectral information (LSI) where GPS signals are unreliable [18] in smart buildings (see Figure 1 Left). Moreover, **outdoor solar forecasting** [20] for predicting future power generation of solar panels on rooftops based on weather

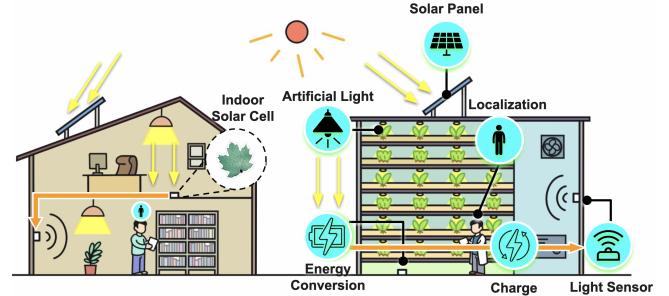


Figure 1: Illustration of PLS tasks in the smart building system (left) [44], and vertical farming system (right) [6]: they are equipped with outdoor solar panels to harness natural energy, self-sustaining sensors is deployed for localization task and powered by indoor decorative solar cells.

conditions and sunlight availability and **indoor solar estimation** for predicting the potential solar output at different locations within an indoor space [7], such as a vertical farm (see Figure 1 Right) [6], are becoming critical as buildings and smart devices shift towards sustainability.

Traditional approaches to solving PLS tasks rely on task-specific deep learning models tailored to each individual application [56]. These models require separate training, manual tuning, and frequent retraining to adapt to changing data distributions, leading to significant engineering overhead. Furthermore, they often struggle to generalize to unseen data or new environments [36], which is particularly problematic in real-world scenarios where domain shifts due to changing conditions are common. Techniques like data augmentation [15] and transfer learning [57] are often employed to mitigate this issue. However, generating high-quality synthetic data is resource-intensive, and transfer learning is limited when the target task diverges significantly from the source domain. Consequently, these models remain both

computationally expensive and inflexible, restricting their scalability across diverse PLS tasks.

Large Language Models (LLMs), with their vast pre-trained knowledge and cross-domain generalization capabilities, present new opportunities to address the challenges outlined above. Inspired by the success of models like ChatGPT [33], which have demonstrated exceptional performance in reasoning and answering complex questions across diverse contexts in natural language processing, LLMs show promise for offering a unified solution across various IoT applications, including light sensing tasks.

However, simply using LLMs with textual prompts, as explored in recent studies [22, 52], is insufficient for these tasks. LLMs are not inherently designed to process sensor data, such as light spectral information or time-series solar output, nor can they generate non-text outputs required for specialized applications like location classification or energy prediction. While recent efforts have extended LLMs beyond traditional language tasks—such as NetLLM [49] for networking and MentalLLM [51] for mental health assessment—these approaches still lack the ability to capture the domain-specific spatial, spectral, and environmental nuances critical for effective PLS sensing across different scenarios and environments.

We propose LightLLM, an innovative model designed to tackle light-based sensing tasks by leveraging the strengths of a pre-trained LLM. It integrates three key components: a task-specific sensor data encoder that extracts meaningful features from sensor inputs, a task-specific natural language prompt that provides contextual information about the environment, and a fusion layer that combines both sensor features and contextual prompts into a unified feature vector. This fused representation is then processed by the pre-trained LLM, which remains entirely frozen but fine-tuned using the LoRA (Low-Rank Adaptation) method. This unique architecture allows LightLLM to efficiently utilize the pre-existing general knowledge of LLMs while seamlessly integrating domain-specific sensor data, enabling it to handle specialized light sensing tasks with high adaptability and minimal retraining effort.

Key contributions of this paper can be summarized as follows:

- We present LightLLM, a general-purpose framework for adapting LLM to a wide range of light-based tasks with minimal effort. To our knowledge, this is the first attempt to use LLM for PLS.
- We conduct comprehensive real-world experiments to validate the effectiveness of LightLLM for three distinct PLS tasks, light-based indoor localization, outdoor solar forecasting, and indoor solar estimation. Our results show the superior performance of

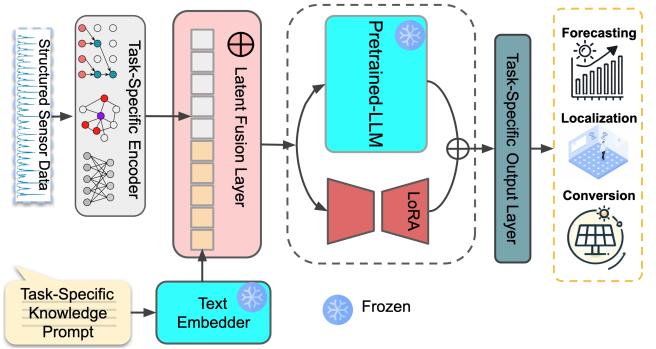


Figure 2: Architecture of LightLLM.

LightLLM in terms of accuracy and generalization compared to state-of-the-art methods.

- We provide an extensive ablation study to evaluate the impact of each component within LightLLM, offering insights into its adaptability and performance across different tasks.

2 Methodology

2.1 System Overview

Our proposed framework, **LightLLM**, integrates advanced encoders for various data modalities with the capabilities of LLMs to address the challenges of diverse tasks. The framework, as illustrated in Figure 2, comprises five key components as followed.

- **Task-Specific Encoders:** Tailored to individual tasks, such as Graph Neural Networks (GNNs) [50] for localization, Temporal Convolutional Networks (TCNs) [25] for forecasting, and Convolutional Neural Networks (CNNs) [27] for energy estimation. For tasks like indoor localization, a custom Knowledge Graph (KG) is integrated to represent a richer context, enabling improved interpretation of sensor data.
- **Task-Specific Knowledge Prompts:** To provide structured contextual information, LightLLM employs task-specific prompts that encode domain knowledge essential to each task. These prompts are embedded as textual inputs, aligning sensor data properties and environmental factors with the LLM’s knowledge base.
- **Latent Fusion Layer (LFL):** A specialized component that integrates the encoded data with latent representations from the LLM. Task-specific prompts are leveraged to enhance the knowledge representation of sensor data.
- **LoRA:** An efficient fine-tuning mechanism that adapts the LLM to specific tasks with minimal computational overhead.

- **Output Projection Layer:** The final layer that projects the combined representations into the desired output format, ensuring compatibility with task-specific requirements.

2.2 Knowledge Graph for Enhanced Data Representation

In LightLLM, the custom KG is designed to capture complex, spatially dependent relationships that are critical for tasks such as LSI indoor localization. By focusing on relevant sensor interactions, the KG enhances the interpretability and efficiency of GNN-based encoders by concentrating on task-specific correlations.

KGs have demonstrated their effectiveness in representing structured data across various domains. Recent research highlights their ability to improve task performance by organizing information into a structured, relational format [8]. In the context of smart building systems, standard ontologies like Brick [5] provide a schema for smart building metadata but fall short in addressing the specific needs of tasks such as LSI indoor localization. For example, such tasks require detailed spatial relationships informed by sensor orientation, light angles, and the presence of obstacles. To bridge this gap, we design a custom KG that captures these unique dependencies, enabling enhanced task performance.

The KG consists of nodes representing sensors and light sources, each with properties such as coordinates (x, y, z), orientation, and detection range. The detailed *Node Connection Algorithm* is outlined in Algorithm 1, which constructs a KG by identifying meaningful connections between sensors and light sources based on spatial criteria. It consists of two phases as follows. In phase 1, for each pair of sensors (s_i, s_j) (Line 2), the Euclidean distance is computed (Line 3). If the distance is within $distance_threshold$, their fields of view (FOVs) are calculated (Lines 5–6). A “correlated” edge is added if their FOVs overlap (Line 7) and no obstacles block the path (Line 8). In phase 2, for each sensor s_i and light source l_k (Line 10), the horizontal distance (d_h) and vertical difference (d_v) are calculated (Lines 11–12). An edge is created if d_h and d_v are less than or equal to the thresholds (Line 13) and no obstacles exist (Line 14).

Figure 3 illustrates an example KG: Sensor 3 is influenced by both Light Sources 1 and 2, while Sensors 1 and 2 are influenced solely by Light source 1. Since Sensors 2 and 3 are spatially correlated, any change in illumination—such as movement by occupants—is likely to affect readings from both sensors. An obstacle positioned between Sensor 1 and Sensors 2/3 blocks any direct correlation between them.

Our KG includes obstacle-free connections only, and focuses on meaningful connections to reduce complexity. Furthermore, it enables dynamic interactions based on FOV and

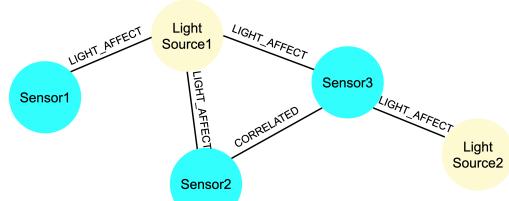


Figure 3: Example of a KG illustrating sensor and light source interactions.

spatial criteria to ensure realistic modeling. This approach ensures the KG effectively captures relationships critical for enhancing LightLLM’s interpretation of sensor data.

Algorithm 1: Node Connection Algorithm

Data: Set of sensors $S = \{s_1, s_2, \dots, s_n\}$, set of light sources $L = \{l_1, l_2, \dots, l_m\}$, set of obstacles O , field of view angle $angle$, distance threshold $distance_threshold$, detection range $range$, vertical threshold $vertical_threshold$

Result: Set of connections between sensors and light sources

```

1  connections ← {} ;
2  for each pair of sensors ( $s_i, s_j$ ) where  $i \neq j$  do
3     $d \leftarrow Distance(s_i, s_j)$  ;
4    if  $d \leq distance\_threshold$  then
5       $fov_i \leftarrow CalculateFOV(s_i, angle, range)$  ;
6       $fov_j \leftarrow CalculateFOV(s_j, angle, range)$  ;
7      if  $FOVOOverlap(fov_i, fov_j)$  then
8        if  $NoObstacle(s_i, s_j, O)$  then
9          Append ( $s_i, s_j, CORRELATED$ ) to
10         connections ;
11
12  for each sensor  $s_i$  and light source  $l_k$  do
13     $d_h \leftarrow HorizontalDistance(s_i, l_k)$  ;
14     $d_v \leftarrow VerticalDifference(s_i, l_k)$  ;
15    if  $d_h \leq distance\_threshold$  and
16     $d_v \leq vertical\_threshold$  then
17      if  $NoObstacle(s_i, l_k, O)$  then
18        Append ( $s_i, l_k, LIGHT\_AFFECTS$ ) to
19        connections ;
20
21 return connections ;
  
```

2.3 Task-Specific Encoders

The task-specific encoders in LightLLM are designed to extract meaningful representations from diverse types of input data, ensuring that the most appropriate encoding techniques are applied.

2.3.1 Localization Encoder: Graph Neural Network For indoor localization, we employ a GNN to interpret spatial relationships among light sensors. The GNN leverages a graph

structure where nodes represent sensor readings, and edges represent spatial relationships, as detailed in Section 2.2. By aggregating information from neighboring nodes, the GNN captures inter-sensor dependencies, enhancing spatial awareness and improving localization accuracy.

2.3.2 Forecasting Encoder: Temporal Convolutional Network For the task of **solar energy forecasting**, we employ a TCN to handle the time-series data from outdoor solar cells. TCNs are ideal for capturing long-range dependencies in time-series data due to their architecture, which combines causal convolutions and dilation.

2.3.3 Energy Estimation Encoder: Convolutional Neural Network To estimate **indoor solar energy output**, we use a CNN to analyze spectral sensor data. The CNN extracts hierarchical features across different wavelengths, helping identify both low-level and high-level patterns in the spectral data. This capability is critical for modeling the complex interactions between light spectra and solar cell energy output.

2.4 Task-Specific Knowledge Prompts

For each task, a **task-specific prompt** is generated to provide additional contextual knowledge. These prompts embed domain-specific features such as spatial information for localization or environmental conditions for solar forecasting. This added context helps guide the model in understanding the relationship between input features.

As shown in Figure 4, each task-specific prompt follows a structured template, which includes:

- Dataset Description: This section outlines the key characteristics of the dataset being used, such as the type of data (e.g., spectral wavelengths) and any relevant factors.
- Task Description: A brief explanation of the task.
- Data Organization: Details on how the data is structured, which helps the model understand the format of inputs.
- Key Input Characteristics: This includes important statistical properties or patterns in the data, such as temporal dependencies or spatial information, that are essential for the task.

It ensures that LightLLM leverages domain knowledge tailored to the unique challenges of each task. For example, in unseen environments where sensor layouts or lighting conditions might differ from the training set, the pre-trained LLM already possesses generalized knowledge about spatial relationships and environmental patterns. When combined with the specific spatial configurations provided in the prompt, LightLLM can better understand how a sensor’s position or a light source’s distribution influences the readings, helping the inference of model.

These prompts are then embedded using the LLM’s text embedding mechanism, and the resulting embedding objects are integrated into the Latent Fusion Layer.

<Dataset Description>: The Power Output Measurements dataset contains data from a 30-kW rooftop photovoltaic array, with power output logged every 1 minute. The dataset captures the power generation patterns under varying sunlight and weather conditions.

<Task Description>: Forecast the next $\{T\}$ steps using the previous $\{P\}$ steps of historical data, accounting for variations in sunlight intensity and environmental factors.

<Data Organization>: $\{PV_output\}$.

<Key Input Characteristics>:

- Sunlight intensity over the past period, $\{min_values\}$ to $\{max_values\}$,
- Cloud cover and weather variability, impacting solar output trends,
- Temporal patterns such as day-night cycles,
- Geographic influence from location-specific sunlight exposure at $\{latitude\}$, $\{longitude\}$.

The focus is on modeling how geographic and environmental factors drive solar output variability for more accurate predictions.

Figure 4: Task-Specific Knowledge Prompt Example: Solar Energy Forecasting Prompt.

2.5 Adaptive Latent Fusion Layer

The LFL in LightLLM is a key component designed to integrate encoded features from task-specific encoders with prompt embeddings generated from task-specific knowledge prompts. This integration is achieved through a multi-head attention mechanism [42], allowing the model to leverage the strengths of both the feature encoders and the LLM, the structure is shown in Figure 5. The LFL ensures that the encoded data is aligned with the LLM’s latent space, facilitating effective fusion of multimodal information.

The encoded features F_{enc} from the task-specific encoder and the embedded prompt P_{embed} are fused using a multi-head attention mechanism within the LFL. Specifically, the query Q , key K , and value V matrices are computed from the encoded features and the prompt embeddings as follows:

$$Q = W_Q F_{enc}, \quad K = W_K P_{embed}, \quad V = W_V P_{embed}.$$

Here, the prompt embeddings P_{embed} encapsulate structured, task-specific knowledge that complements the encoded sensor features. By assigning P_{embed} as both the key and value, the attention mechanism identifies relevant knowledge and integrates it into the encoded features F_{enc} . The

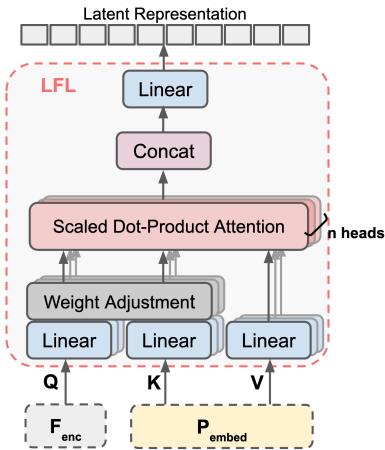


Figure 5: Overview of Latent Fusion Layer.

key (K) assesses relevance, while the value (V) provides the information to enhance the final representation.

Each head in the multi-head attention mechanism works with different learned weight matrices W_Q, W_K and W_V . It splits the matrices into n_{heads} parallel subspaces, computes attention separately for each head, and then concatenates the results.

In addition to the standard attention mechanism, the LFL incorporates a weight adjustment mechanism to control the contribution of the encoded data and the text embeddings. This is achieved through learnable parameters α and β , which adjust the weights of the target and source embeddings, respectively. The target embedding T and source embedding S are projected and scaled as follows:

$$T' = \alpha \cdot Q, \quad S' = \beta \cdot K, \quad (1)$$

where T' and S' are the weighted target and source embeddings. The attention scores are then computed using the scaled dot-product of the weighted embeddings:

$$\text{Scores} = \text{softmax} \left(\frac{T' S'^T}{\sqrt{d_k}} \right), \quad (2)$$

where d_k is the dimension of the key vectors. Once the fusion process is completed, the resulting representation is passed into the LLM to generate the reprogrammed latent representation. The resulting latent representation combines the strengths of the encoded sensor data and the prompt knowledge, providing a task-specific feature that is well-suited for downstream tasks in PLS.

The LFL also includes normalization and dropout layers to ensure stable training and prevent overfitting. LFL effectively leverages both pre-trained knowledge and task-specific information, allowing LightLLM to produce highly accurate, adaptable outputs across various PLS applications.

2.6 Low-Rank Adaptation

To efficiently adapt the pre-trained language model for our specific downstream tasks, we utilize LoRA [17], which introduces low-rank matrices around the original pre-trained LLM, allowing for dimensionality reduction and expansion, effectively performing a rank adaptation. During training, the parameters of the LLM remain frozen, while only the low-rank matrices A and B are trained. This significantly reduces the number of trainable parameters while preserving the model's capacity to adapt to new tasks.

Initially, the reduction matrix A is randomly initialized using a Gaussian distribution, while the expansion matrix B is initialized as a zero matrix. This setup ensures that the pre-trained behavior of the LLM is maintained at the beginning of training, with the adaptation gradually introduced as the matrices are trained.

The modified weight matrix W' is computed as follows:

$$W' = W + \alpha AB, \quad (3)$$

where W is the pre-trained weight matrix (e.g., a frozen LLM). The dimensions of low-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ are significantly smaller than the full weight matrix $W \in \mathbb{R}^{d \times d}$, significantly reducing the number of parameters being fine-tuned.

During the forward pass, the input x is computed as follows:

$$W'x = Wx + \alpha(AB)x. \quad (4)$$

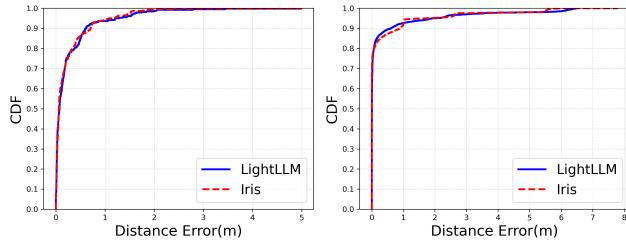
This approach is analogous to a residual connection, where the low-rank updates simulate the full fine-tuning process by introducing an adaptable layer on top of the pre-trained LLM. When the rank r equals the dimension of W , LoRA effectively performs full fine-tuning, but with $r \ll d$, LoRA provides a more computationally efficient way to adapt the model. In the inference phase, LoRA introduces minimal additional latency, with parallel computations for Wx and $(AB)x$.

In our model, we empirically chose the settings $r = 64$, $\alpha = 64$, and dropout of 0.1, to achieve an optimal balance between task-specific adaptation and preserving the pre-trained knowledge of the LLM, while maintaining computational efficiency.

2.7 Task-Specific Output Layer

The output layer of LightLLM is designed to handle the specific requirements of different tasks by employing task-specific heads. This ensures that the final model outputs are appropriate for each application.

For example, for the localization task, the output layer is configured as a probability classification head [14]. This head



(a) Apartment

(b) Office

Figure 6: Localization error under *seen* scenarios.

outputs a probability distribution over possible locations, allowing the model to predict the most likely position based on the input data. The classification head uses a softmax activation function to convert the model’s logits into probabilities, thereby facilitating accurate location predictions.

The flexible design of this task-specific head allows LightLLM to be easily adapted to different tasks. By incorporating the appropriate headers, the model is able to provide accurate, task-relevant output, whether it involves classification, regression, or other output formats. This module ensures that the model’s predictions are not only accurate but also customized to the unique characteristics of each PLS task.

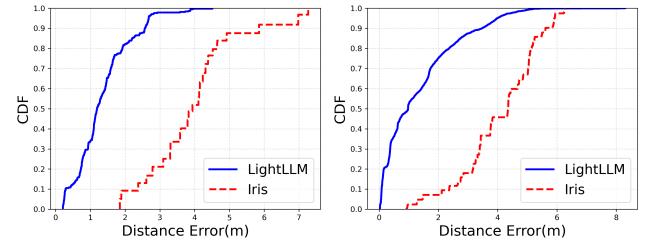
3 EVALUATION

3.1 Light Spectral Localization

Indoor localization using light spectral information has proven to be a highly effective technique for achieving passive Visible Light Positioning (VLP) in indoor environments. This method leverages the unique properties of light in different spaces to estimate the location of users. One of the leading systems for spectral-based localization is *Iris* [18], which employs a conditional Generative Adversarial Network (cGAN) for data augmentation to enhance its performance.

To evaluate the performance of LightLLM in indoor localization, we collected data from two distinct indoor environments: a large office and a apartment. In the office environment, we deployed 27 spectral sensors placed strategically throughout the 108 square meter area. In the apartment environment, we deployed 17 spectral sensors across the 25 square meter space. Each environment was carefully configured to capture spectral data over multiple sessions.

In the **seen environment** tests, where both training and testing data are collected from the same indoor room, LightLLM shows comparable or slightly better performance than *Iris* (see Figure 6). For example, in the 90th percentile error, where LightLLM achieves 0.608m vs 0.705m for *Iris* in the apartment environment, and LightLLM achieves 0.89m,



(a) Apartment

(b) Office

Figure 7: Localization error under *unseen* scenarios.

while *Iris* performs slightly worse at 0.60m of 90th percentile error in the office environment.

In the **unseen environment**, where the training data does not include the environment used in the test set (e.g., train with apartment’s dataset and test with office’s dataset). This setting could evaluate how well the system generalizes to new environments, which is a significant challenge for traditional models like *Iris*.

The results in Figure 7 show that *Iris* sees a dramatic increase in error in both environments, with a median error of 3.93m and 4.35m, respectively, a significant degradation from its seen performance. In contrast, LightLLM maintains far better generalization with a median error of only 0.98m in office and 1.19m in apartment, achieving 4.4x improvement. Similarly, 75th percentile and 90th percentile error further emphasize this difference, for example, LightLLM achieves 1.99m and 3.46m compared to 5.05m and 5.61m for *Iris* in the office. This demonstrates LightLLM’s superior ability to handle difficult and unseen localization tasks, where traditional method failed to work.

In summary, LightLLM exhibits competitive performance in seen environments but shows superior generalization capabilities in unseen environments. The ability of LightLLM to adapt to new environments with minimal degradation in performance highlights its advantage over traditional systems like *Iris*, which almost impossible to work in a completely new environment with unfamiliar data. This makes LightLLM a more robust and scalable solution for light spectral indoor localization, especially in real-world applications where environments are diverse and complex.

3.2 Outdoor Solar Energy Forecasting

Solar forecasting plays a crucial role in optimizing energy management systems and improving the reliability of renewable energy sources. Accurate forecasts of photovoltaic (PV) generation help ensure energy grid stability and enable more efficient integration of solar energy into the energy mix. To achieve these predictions, solar forecasting models typically rely on a variety of data models, including historical PV generation and environmental data, such as images of the sky that capture cloud movement.

The dataset used in this evaluation, SKIPP'D [31] includes over three years of PV power generation data. It covers a diverse range of weather conditions, providing an excellent opportunity to assess the performance of solar forecasting models across various scenarios. The dataset includes PV output data recorded at high temporal resolution, offering rich insights into power generation patterns over time.

To evaluate the performance of LightLLM, we employ several commonly used metrics in solar forecasting [13]:

- **Continuous Ranked Probability Score (CRPS):** A lower CRPS indicates more accurate probabilistic predictions.
- **Forecast Skill (FS):** This metric reflects the percentage improvement over a baseline method (e.g., smart persistence [9]), with higher FS values representing better improvements.
- **Winkler Score (WS):** This metric evaluates the width of the prediction intervals at a given probability level, typically aiming for narrow intervals that still capture the true value. The lower WS indicates a better probabilistic forecast.

These metrics enable a thorough assessment of how well different forecasting models predict solar energy output under various conditions.

We compared LightLLM against several state-of-the-art methods, including traditional method ConvLSTM [39], and SkyGPT [32], a leading model that employs sky images in combination with U-Net for solar energy forecasting, and TimeLLM [23], which also utilizes LLM for time-series predictions. All of them are comparing with a commonly used baseline method, smart persistence model [9]. The results in the seen environment are shown in the Table 1. LightLLM outperforms SkyGPT in several key metrics, obtaining a CRPS of 1.92 kW, representing a 33.7% improvement in FS, which is significantly higher than SkyGPT's performance. Furthermore, LightLLM records a WS of 20.80, demonstrating superior accuracy in matching the predicted solar output distribution to the actual data. Additionally, while TimeLLM offers solid performance, LightLLM consistently delivers better results across all metrics, highlighting the advantage of integrating task-specific encoders and the LFL in LightLLM.

In the unseen scenario, LightLLM is evaluated on a completely separate time period of data that the model has not encountered during training, specifically, we trained by using the data from 2017 Mar to 2019 Oct and tested by using data from 2019 Nov to 2019 Dec. It is intended to simulate real-world conditions to evaluate the final performance of the model. By looking at the results shown in Table 1 and Figure 8, LightLLM maintains its edge with a CRPS of 2.52 kW and a FS improvement of 31.4%, outperforming SkyGPT (CRPS: 2.81 kW, FS: 23.4%) and TimeLLM (CRPS: 2.67 kW, FS:

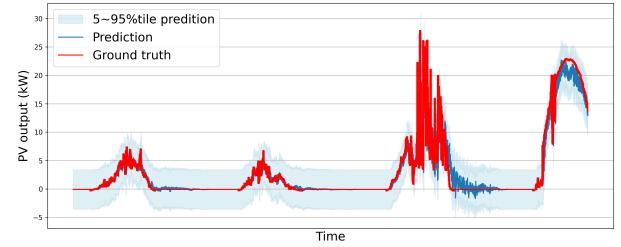


Figure 8: Prediction curve of LightLLM and ground truth with prediction intervals (5-95%) under unseen scenarios.

27.3%). Furthermore, LightLLM records a WS of 22.56, again lower than SkyGPT's 26.70 and TimeLLM's 24.30, demonstrating its robustness. This indicates that LightLLM not only excels in seen conditions but also generalizes well in unseen settings.

The comparison clearly shows that LightLLM offers better performance across both seen and unseen environments, providing superior accuracy and generalization capabilities compared to other leading models such as SkyGPT and TimeLLM.

3.3 Indoor Solar Energy Estimation

3.3.1 Challenges in Real-World Solar Energy Estimation. With the increasing deployment of solar cells for indoor energy harvesting applications, such as those used in low-power IoT devices [1] or used as interactive surfaces [30], presents an opportunity to harness light more effectively for energy generation in indoor spaces. Given that different types of solar cells exhibit unique responses to varying light conditions, optimizing their placement and understanding their energy output in real-world environments becomes critical. This optimization can greatly enhance the efficiency of solar energy systems in indoor settings.

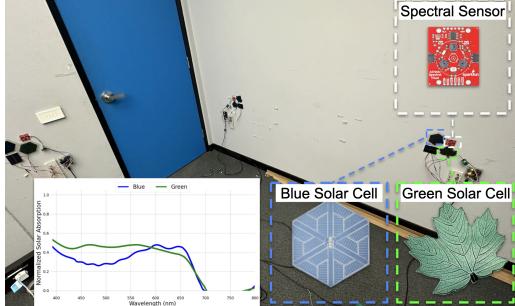
Fortunately, existing technology allows us to simulate and collect spectral sensor data [19]. However, the challenge lies in accurately translating this spectral data into solar energy output for each unique solar cell. While mathematical models exist to predict this conversion under idealized conditions, their accuracy diminishes in complex, real-world scenarios due to various discrepancies between theory and practice.

In theory, the short-circuit current density J_{sc} is a parameter in assessing the performance of a solar cell. As described in [48], this value depends on the external quantum efficiency (EQE), which is the proportion of photo-generated electrons to incident photons, and the intensity of photons at each wavelength $P_{in}(\lambda)$. Following the approach in [29], the total current generated by a solar cell can be expressed as:

$$\text{Current} = A \cdot \int_{\lambda_{min}}^{\lambda_{max}} a(\lambda) \cdot I(\lambda) \cdot \lambda \cdot d\lambda, \quad (5)$$

Table 1: LightLLM vs. state-of-the-art for outdoor solar forecasting, with Smart Persistence as the baseline.

Method	Seen			Unseen		
	CRPS[kW]↓	FS[%]↑	WS↓	CRPS[kW]↓	FS[%]↑	WS↓
Smart Persistence	2.89	0	-	3.67	0	-
ConvLSTM	2.70	6.6	41.12	3.09	15.8	43.82
SkyGPT	2.49	13.8	26.72	2.81	23.4	26.70
TimeLLM	2.27	21.6	21.30	2.67	27.3	24.30
LightLLM	1.92	33.7	20.80	2.52	31.4	22.56

**Figure 9: Environment for deploying spectral sensors and semi-transparent solar cells with different absorption rates.**

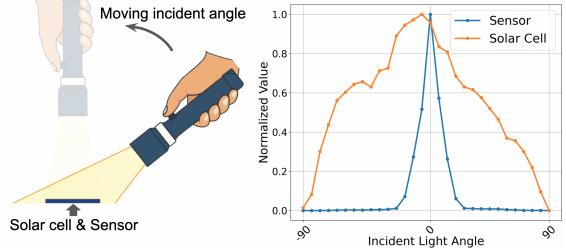
given the spectral absorption rate $a(\lambda)$, the incident spectrum $I(\lambda)$, and the surface area A of the solar cell.

Table 2: Actual vs. predicted photocurrent (mA) for solar cells at various indoor locations.

Location	Blue Solar Cell		Green Solar Cell	
	Actual	Predicted	Actual	Predicted
1	0.596	0.335	0.459	0.249
2	0.547	0.306	0.523	0.242
3	0.097	0.141	0.348	0.517
Avg MAPE[%]	44.31		49.34	

However, our real-world experiments revealed significant discrepancies between theoretical predictions and actual solar cell performance. In these tests, we deployed both spectral sensors (AS7265x) with 18 wavelength channels and solar cells with different absorption rates in an indoor room setup, as shown in Figure 9. Each solar cell's and spectral sensor's output was measured using the same Arduino device and synchronized to ensure consistency. We compared the measured outputs with predicted values from Eq. (5) and used Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) as the evaluation metrics, calculated as follows:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\text{Actual}_i - \text{Predicted}_i}{\text{Actual}_i} \right| \times 100, \quad (6)$$

**Figure 10: Experiment results show the difference between the solar cell and light sensor readings when the light hits them from different angles.**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\text{Actual}_i - \text{Predicted}_i)^2, \quad (7)$$

where n represents the total number of data samples, with $n = 1,035$ in our experiments.

From the results of Table 2, we observed an average MAPE of **46.83%** across different room configurations, while the individual MAPE values for each solar cell are as follows: Solar Cell Blue **44.31%** and Solar Cell Green **49.34%**. This high error rate underscores the limitations of the equation-based approach.

One of the fundamental reasons we found during our experiments was the discrepancy between the behavior of spectral sensors and solar cells. Spectral sensors [4] are designed with a defined FOV. In contrast, solar cells are not optimized for measuring light at specific angles but aim to maximize overall absorption across their surface area [35]. Solar cells perform most efficiently when light is incident perpendicular to the surface. However, as the angle of incidence deviates from the vertical axis, the efficiency of light absorption may decrease. To investigate this, we conducted experiments where both spectral sensors and solar cells were deployed at the same location and under different angles of light incidence, the result is shown in Figure 10.

Additionally, we observed several other factors that contributed to inaccuracies when applying the mathematical models to real-world scenarios. These factors include variations in the packaging of solar cells, which can affect the way light interacts with the cell surface, as well as manufacturing inconsistencies in the absorption curve data provided by the

Table 3: Comparison of different machine learning methods vs. LightLLM for indoor solar estimation.

Method	Seen		Unseen	
	MSE	MAPE[%]	MSE	MAPE[%]
Linear Regression	19.40	6.15	3592.97	168.26
Random Forest	5.76	2.58	2438.28	140.91
Gradient Boosting	5.50	2.62	2465.54	141.64
Decision Tree	10.29	2.70	2934.94	154.20
SVR	15.96	5.41	2013.49	128.54
MLP	6.11	2.68	819.63	83.02
CNN	5.98	2.60	959.81	89.61
LightLLM	5.72	2.59	263.21	26.33

solar cell manufacturers. These discrepancies make it clear that relying purely on physical equations to estimate indoor solar energy is not feasible in complex environments.

Given the limitations of theoretical models in predicting solar cell performance under real-world conditions, we turn to data-driven approaches.

3.3.2 Performance Evaluation Results Under Seen Environment. We then tested several machine learning methods under a seen environment, where the training set contained data from the same environment as the test set. As summarized in Table 3. In this controlled setting, data-driven methods significantly outperformed the physical equation-based approach. Among the data-driven methods, Random Forest achieved the lowest MAPE of 2.58%, while LightLLM closely followed with a MAPE of 2.59%. In terms of MSE, LightLLM achieved the best performance 5.72. This demonstrates that, given sufficient training data from the same environment, the data-driven models can effectively capture the relationship between spectral data and solar cell energy output.

Results Under Unseen Environment. To further test the generalization of the models, we evaluated them on unseen environments, where the test set was drawn from environments not present in the training set. We specifically altered the lighting and spatial configuration within the room. For example, we introduced two diffused standing lamps, and rearranged the room layout by changing the furniture. The results for this setting are shown in Table 3. Most machine learning models suffered a significant performance drop. For example, Linear Regression and Random Forest showed MAPE values exceeding 140%, illustrating their tendency to overfit and their lack of generalization capability. In contrast, LightLLM demonstrated remarkable robustness, achieving a MAPE of 26.33% and MSE of 263.21, the best among the compared methods. We note that LightLLM consistently outperforms the traditional equation-based approach in both seen and unseen environments.

3.4 Ablation Study

3.4.1 LightLLM vs. Direct Prompting To evaluate the effectiveness of LightLLM against a simpler method of directly prompting an LLM (e.g., ChatGPT-4), we firstly conducted an experiment using the indoor localization task as a representative. We compared both zero-shot and few-shot prompting approaches using models like GPT-4, GPT-3.5 [33], Llama 3 [41], and Mistral Large 2 [3]. Our goal was to determine whether carefully crafted prompts, including chain-of-thought (CoT) [45] prompting strategy, could compete with the task-specific design of LightLLM.

Prompting Setup. The ability of LLMs to understand and reason across a wide range of tasks makes them powerful, particularly when paired with well-constructed prompts. Recent studies [28] have demonstrated that proper prompt engineering can significantly improve LLM performance on domain-specific tasks. Therefore, for the direct prompting method, we created a detailed prompt, leveraging CoT reasoning to improve its ability to process the sensor data step by step. A prompt example design is shown as in Figure 11. First, the **Instruction** gives a detailed description of the room setup, including the locations and orientations of spectral sensors, which is essential for the model to understand the spatial relationships in the environment. The **Data** section introduces the sensor readings and the changes in wavelength due to movement. The **Question** then prompts the LLM to determine the person’s location based on the given data, while the inclusion of a step-by-step analysis encourages logical reasoning, following the CoT prompting strategy.

Instruction: In an indoor room, illuminated by overhead lights, we have placed spectral sensors all around, facing inside the room.

The location of spectral sensors are as follows: $\{\text{sensor locations } (x, y, z)\}$. Their orientation are as following: $\{\text{sensor orientations } (x, y, z)\}$.

When someone passes by, the light changes, hence the readings on each wavelength of the spectral sensors also change. When the room is empty, the sensors’ readings are as follows: $\{\text{readings}\}$.

Data: The sensor data given in the room has 18 wavelengths. The change of 18 wavelengths’ reading comparing with the empty room for each sensor are given below: $\{\text{changes}\}$.

Question: The person’s location belongs to one of the following categories: $\{\text{locations } (x, y)\}$. Please tell me where the person is based on the given information. Make an analysis step by step.

Response: $\{\text{answer}\}$

Figure 11: Prompt example for LSI-based localization.

Table 4: Performance comparison of direct prompting vs. LightLLM across models and tasks.

Method	Model	Outdoor Solar Forecasting	Indoor Localization	Indoor Solar Estimation
		CRPS[kW]↓	Median Error[m]↓	MAPE[%]↓
Zero-shot Prompting	<i>GPT-4</i>	2.57	1.77	48.72
	<i>GPT-3.5</i>	2.96	2.03	45.35
	<i>Llama 3</i>	3.37	3.84	45.44
	<i>Mistral Large 2</i>	2.69	2.36	56.38
Few-shot Prompting	<i>GPT-4</i>	2.42	1.70	50.43
	<i>GPT-3.5</i>	2.87	1.92	45.04
	<i>Llama 3</i>	3.20	3.12	44.80
	<i>Mistral Large 2</i>	2.55	2.30	49.06
LightLLM	-	1.64	0.98	26.33

Zero-shot prompting. As shown in Table 4, GPT-4 achieved the best performance among different models, a median error of 1.77m for the LSI localization task. This error is considerably larger than the result achieved by LightLLM in the same scenario, where the median error was 0.98m. From the responses given by ChatGPT-4, we also found that, despite the detailed instruction and CoT prompt, ChatGPT-4 primarily used a very basic heuristic approach: it attempted to estimate the person’s location by selecting the coordinates closest to the sensors that detected the largest change in spectral readings.

To inject domain-specific context without additional training, researchers have explored **Few-shot prompting** by embedding a few examples directly within the prompt (e.g., [2]) while the LLM parameters without tuning. From the results in Table 4 we can observe, while this did improve the performance slightly (e.g., median error from 1.77m to 1.70m), the results still fell short compared to LightLLM.

Furthermore, we observed that certain models, such as Gemini 1.5 [38], refused to provide specific answers to queries about location coordinates. Instead, it offered approximate descriptions, such as “The estimated location based on the given sensor data is near the bottom-center of the room,” or stated, “I cannot provide a specific coordinate.”

The limitations of direct prompting were not confined to localization. We have also conducted the experiment for the other two tasks. We used a smaller scale of the dataset to evaluate the performance of solar energy forecasting. The forecasting results by direct prompting were poor, leading to high CRPS values (e.g., GPT-4 zero-shot at 2.57 kW, compared to LightLLM’s 1.64 kW). Similarly, for indoor solar energy estimation, the prompt-based estimates deviated significantly from actual measurements, as the LLMs failed to model the complexities of spectral data conversion (e.g., Llama 3 at 45.44% vs LightLLM’s 26.33%). Interestingly, when we used few-shot prompting by providing some examples, the performance of certain models decreased slightly in indoor solar

estimation task. This decline can be attributed to the fact that these examples may have misled the LLM models, causing them to overfit to specific patterns.

This ablation study shows that, LightLLM was able to handle the complexity of the sensor data much more effectively, demonstrating that for tasks requiring complex multimodal data interpretation, it is not sufficient to simply use LLM with direct prompting methods.

3.4.2 Impact of Prompt Knowledge and LFL The inclusion of prompt knowledge and the LFL plays a important role in enhancing LightLLM’s performance, particularly in unseen scenarios where the system must generalize to new environments not encountered during training. To demonstrate the importance of these components, we compare the performance of LightLLM with and without the prompt knowledge and LFL.

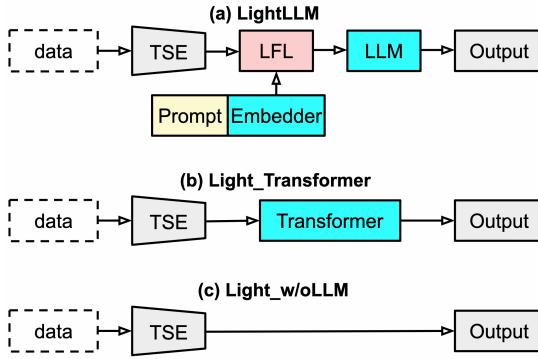
Among the tasks evaluated, localization shows the most notable improvement with prompt knowledge and LFL, in the unseen scenario, LightLLM with these components achieves a 90th percentile error of 3.46 meters, a considerable improvement over the 5.41 meters when they are removed. Table 5 also present the results for other two tasks. For indoor solar estimation, a similar trend can be observed. The system with prompt knowledge, which includes detailed descriptions of solar cell placement and light conditions, reduces the prediction error, achieving a MAPE of 26.33%, compared to 35.42% without those. Similarly, we could also observe an improvement in the solar forecasting task, CRPS decrease from 2.80kW to 2.52kW.

3.4.3 Impact of Task-specific Encoder and Low-Rank Adaptation Both task-specific encoders (TSE) and LoRA are essential in affecting the performance of LightLLM across various tasks. To assess their impact, we evaluated the model’s performance with and without these components.

The results are shown in Table 5. For example, for the task of solar energy estimation, in unseen scenario, LightLLM

Table 5: Impact of LFL, TSE, and LoRA.

Task	Metric	Configuration			
		Normal	w/o LFL	w/o TSE	w/o LoRA
Outdoor Solar Forecasting	CRPS[kW] \downarrow	2.52	2.80	2.83	2.57
Indoor Localization	90th Percentile Error[m] \downarrow	3.46	5.41	3.73	3.79
Indoor Solar Estimation	MAPE[%] \downarrow	26.33	35.42	30.10	28.45

**Figure 12: Three ablation versions of LightLLM.**

with LoRA reduces the average MAPE from 28.45% to 26.33%. Moreover, for the solar energy forecasting task, the inclusion of a task-specific encoder brings a 12% improvement in the CRPS metric, demonstrating its ability to capture the temporal dependencies in time-series data.

In summary, both LoRA and task-specific encoders provide considerable performance enhancement, with LoRA ensuring efficient fine-tuning for task-specific adjustments and task-specific encoders capturing the unique characteristics of each task. Together, these components make LightLLM highly accurate, flexible, and scalable across diverse applications and environments.

3.4.4 Impact of Knowledge Graph In this section, we evaluate the contribution of the KG mentioned in Section 2.2. In the ablation setting, we remove the KG creation in LightLLM, and instead, using raw data as input. We evaluated the models based on our LSI Localization task.

In our evaluation results, the model with KG achieved significantly lower error, with a median error of 0.98 meters. In contrast, removing the KG led to a 50% decrease in localization accuracy, with the median error increasing to 1.42 meters. The findings validate the design rationale presented in Section 2.2, demonstrating that the KG’s structured data is instrumental in achieving precise localization within the LightLLM framework.

**Figure 13: Performance comparison of three ablation versions of LightLLM as shown in Figure 12.**

3.4.5 Impact of LLM in LightLLM To assess the importance of the LLM component within LightLLM, we conducted an ablation study by comparing three versions of our model as shown in Figure 12: LightLLM (full version), Light_Transformer (replacing the LLM component with a single transformer), and Light_w/oLLM (completely removing the LLM, using a simplified model). The purpose of this experiment is to evaluate whether the LLM plays a critical role in the performance of LightLLM, particularly for tasks like solar energy forecasting. The results in Figure 13 demonstrated that LightLLM (with the LLM component) consistently outperformed the other two versions across all evaluated metrics. This ablation study thus reinforces the importance of incorporating LLMs in LightLLM.

4 FUTURE DIRECTIONS

Exploration of Different LLM Models: Our next attempt is to understand whether and how different pre-trained LLM models may affect the performance of LightLLM. To this end, we compared the performance of LLaMA-7B, which has 7 billion parameters, against GPT-2 with only 1.5 billion parameters across all tasks. The results in Figure 14 demonstrated that LLaMA-7B consistently outperformed GPT-2.

The improvements in performance using LLaMA-7B can be attributed to its larger model size and more recent training, which have equipped it with a more extensive knowledge base and greater capacity to model complex patterns, enabling better reasoning and adaptation for spectral and temporal data.

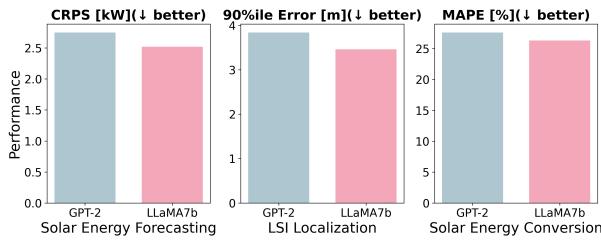


Figure 14: Performance comparison between using different LLM models in LightLLM framework.

While our experiments demonstrated the benefits of using larger models, we need to consider the trade-offs between model complexity, performance, and resource demands. Models with more parameters, like GPT-4, may further boost performance, especially for tasks requiring deeper reasoning. Future research could explore different language model architectures to identify those best suited for various PLS tasks and their impact on adaptability in complex environments.

Task-Specific Tuning Overhead: Although LoRA reduces the need for full model retraining, there is still some overhead involved in task-specific tuning. Fine-tuning LLMs for each new task, especially when introducing novel sensor modalities, can require additional data collection and adjustments to components such as the encoder. Future work could explore more efficient methods to streamline these processes, for example, advancing strategies to minimize data requirements and automate adaptation for emerging sensor types could make tuning faster and more resource-efficient.

5 RELATED WORK

5.1 PLS Tasks and Algorithms

The field of PLS tasks encompasses applications like indoor localization, solar energy forecasting, and indoor solar energy estimation. These tasks are critical for optimizing energy systems, enhancing smart environments, and improving the efficiency of renewable energy grids.

Indoor localization using light spectral information, on the other hand, is essential for services like smart navigation and asset tracking in smart buildings. Data collected from light sensors offers a promising alternative to traditional localization methods. Existing models are tailored to specific environments, and it usually relies on fingerprint collection to train the model, which requires extensive data collection and may not generalize well to new spaces [18, 24, 43].

Solar radiation forecasting helps predict future energy output, improving the reliability and efficiency of energy distribution. Recent advancements in this field have largely focused on deep learning-based methods, such as combining CNN with recurrent neural network (RNN) [16, 34, 39, 54].

There is also a growing trend of replacing RNN with transformers to better capture temporal patterns in the data and improve forecasting accuracy [32].

With the development of efficient solar cells for indoor environments [21, 30, 48], there is a growing need for accurate indoor solar energy estimation. This task focuses on accurately predicting the energy output of photovoltaic systems in indoor conditions to maximize their utility. Current approaches often rely on the solar models that have been presented that use mathematical linear [53] and nonlinear functions [12].

Despite the progress made in solving these tasks, several significant challenges remain. One of the primary issues is the lack of generalization across different environments or unseen data [36]. While data augmentation [15, 55] can help improve performance in the short term, it is computationally expensive and resource-intensive, and it does not fundamentally address the issue of generalization across diverse tasks. Another limitation arises from the fixed nature of these models: a model designed for indoor localization cannot be easily repurposed for solar energy forecasting without re-engineering the entire architecture [46]. This rigidity limits the development of a scalable, multi-task solution capable of handling the diversity of data modalities in PLS tasks.

5.2 Large Language Model and Applications

In recent years, LLMs achieved a great success in the field of natural language processing (NLP). Models like BERT [10], GPT [33], and LLaMA [41], which are built on the Transformer architecture, have demonstrated their ability to perform reasoning, translation and text generation tasks. One of the most significant advancements in LLMs is their ability to learn from massive datasets, capturing latent knowledge that allows them to perform well across diverse tasks.

The flexibility of LLMs has made them increasingly attractive for applications beyond NLP. Techniques like LoRA [17] and prefix-tuning [26] have made it possible to fine-tune LLM for specific downstream tasks with minimal computational overhead. This has led to the exploration of LLM in non-textual domains, including cross-modal learning where text is combined with images, audio, or sensor data.

Recent studies have shown the potential of LLM in multimodal tasks, where they are adapted to handle inputs from various sources beyond text. For example, models like CLIP [37] successfully integrate image-text pairs, allowing LLM to perform visual reasoning tasks. Similarly, PaLM-E [11] has been applied to control robotics tasks, showcasing how LLM can extend their capabilities into non-linguistic domains by integrating structured knowledge from other modalities.

In fields such as healthcare, LLM have been adapted to process clinical text alongside sensor data to provide predictions for medical outcomes. For example, medical models like PaLM-2 [40], MentalLLM [51] have shown success in fine-tuning LLM to accurately interpret clinical notes and predict patient outcomes based on multimodal inputs. Additionally, NetLLM [49] adapts LLM for networking tasks such as viewport prediction and cluster job scheduling by leveraging the feature from networking task data, while TimeLLM [23] specializes in time-series data processing, illustrating the growing versatility of LLM across diverse domains. Similarly, FM-Fi [47] explores LLM applications in human activity recognition (HAR) using RF data. These advancements illustrate the cross-domain adaptability of LLMs, inspiring the development of LightLLM for PLS tasks.

6 CONCLUSION

We presented LightLLM, a unified framework designed to address diverse PLS tasks such as indoor localization, outdoor solar forecasting, and indoor solar estimation. The Latent Fusion Layer and task-specific prompts enable effective multimodal sensor fusion, providing a rich context information. Moreover, by integrating LLM with task-specific encoders and utilizing techniques like LoRA for efficient fine-tuning, LightLLM achieved significant improvements in accuracy and adaptability, particularly in unseen scenarios. Our extensive evaluations showed that LightLLM outperforms traditional methods and simple prompting approaches, in handling complex, real-world data. Though evaluated on different tasks, LightLLM's design offers a general-purpose solution for other context-rich sensing applications, establishing a foundation for LLM-based frameworks in PLS.

Acknowledgments

This work was partly supported by the Australian Research Council Discovery Project DP210100904 and CSIRO Data61 PhD Scholarship Program.

References

- [1] 2023. Original Samsung Smart TV Remote (Solar Cell Remote Control - BN59-01385B Genuine). <https://remotecontrolwarehouse.com.au/products/>.
- [2] Monica Agrawal, Stefan Heggemann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. *arXiv preprint arXiv:2205.12689* (2022).
- [3] Mistral AI. 2024. Mistral. <https://mistral.ai/>.
- [4] ams OSRAM. 2022. AS7265x Smart Spectral Sensor. <https://ams.com/en/as7265x>.
- [5] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, et al. 2016. Brick: Towards a unified metadata schema for buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. 41–50.
- [6] Kurt Benke and Bruce Tomkins. 2017. Future food-production systems: vertical farming and controlled-environment agriculture. *Sustainability: Science, Practice and Policy* 13, 1 (2017), 13–26.
- [7] Swarup Biswas and Hyeok Kim. 2020. Solar cells for indoor applications: Progress and development. *Polymers* 12, 6 (2020), 1338.
- [8] Kathrin Blagoe, Adriano Barbosa-Silva, Simon Ott, and Matthias Samwald. 2022. A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks. *Scientific Data* 9, 1 (2022), 322.
- [9] Aldo Vieira Da Rosa and Juan Carlos Ordóñez. 2021. *Fundamentals of renewable energy processes*. Academic Press.
- [10] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [11] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378* (2023).
- [12] AA El-Sebaii, FS Al-Hazmi, AA Al-Ghamdi, and Saud Jameel Yaghmour. 2010. Global, direct and diffuse solar radiation on horizontal and tilted surfaces in Jeddah, Saudi Arabia. *Applied energy* 87, 2 (2010), 568–576.
- [13] Tilmann Gneiting and Matthias Katzfuss. 2014. Probabilistic forecasting. *Annual Review of Statistics and Its Application* 1, 1 (2014), 125–151.
- [14] Ian Goodfellow. 2016. Deep learning.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- [16] Vincent Le Guen and Nicolas Thome. 2020. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11474–11484.
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [18] Jiawei Hu, Yanxiang Wang, Hong Jia, Wen Hu, Mahbub Hassan, Brano Kusy, Ashraf Uddin, and Moustafa Youssef. 2023. Iris: Passive Visible Light Positioning Using Light Spectral Information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023), 1–27.
- [19] Jiawei Hu, Yanxiang Wang, Hong Jia, Cheng Jiang, Mahbub Hassan, Brano Kusy, and Wen Hu. 2024. LiDARSpectra: Synthetic Indoor Spectral Mapping with Low-cost LiDARs. In *2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 75–87.
- [20] Rich H Inman, Hugo TC Pedro, and Carlos FM Coimbra. 2013. Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science* 39, 6 (2013), 535–576.
- [21] Arturo Fajardo Jaimes and Fernando Rangel de Sousa. 2017. Simple modeling of photovoltaic solar cells for indoor harvesting applications. *Solar Energy* 157 (2017), 792–802.
- [22] Sijie Ji, Xinzie Zheng, and Chenshu Wu. 2024. HARGPT: Are LLMs Zero-Shot Human Activity Recognizers? *arXiv preprint arXiv:2403.02727* (2024).
- [23] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).

- [24] Daniel Konings, Nathaniel Faulkner, Fakhruл Alam, Edmund M-K Lai, and Sergo Demidenko. 2019. FieldLight: Device-free indoor human localization using passive visible light positioning and artificial potential fields. *IEEE Sensors Journal* 20, 2 (2019), 1054–1066.
- [25] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 156–165.
- [26] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).
- [27] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* 33, 12 (2021), 6999–7019.
- [28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.
- [29] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, Mushfika B Upama, Ashraf Uddin, and Moustafa Youssef. 2019. Solargest: Ubiquitous and battery-free gesture recognition using solar cells. In *The 25th annual international conference on mobile computing and networking*. 1–15.
- [30] Yogesh Kumar Meena, Krishna Seunarine, Deepak Ranjan Sahoo, Simon Robinson, Jennifer Pearson, Chi Zhang, Matt Carnie, Adam Pockett, Andrew Prescott, Suzanne K Thomas, et al. 2020. PV-tiles: towards closely-coupled photovoltaic and digital materials for useful, beautiful and sustainable interactive surfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [31] Yuhan Nie, Xiatong Li, Andea Scott, Yuchi Sun, Vignesh Venugopal, and Adam Brandt. 2023. SKIPP'D: A SKy Images and Photovoltaic Power Generation Dataset for short-term solar forecasting. *Solar Energy* 255 (2023), 171–179.
- [32] Yuhan Nie, Eric Zelikman, Andea Scott, Quentin Paletta, and Adam Brandt. 2024. SkyGPT: Probabilistic ultra-short-term solar forecasting using synthetic sky images from physics-constrained VideoGPT. *Advances in Applied Energy* 14 (2024), 100172.
- [33] OpenAI. 2023. GPT-4 Technical Report. <https://openai.com/research/gpt-4>.
- [34] Quentin Paletta, Guillaume Arbod, and Joan Lasenby. 2021. Benchmarking of deep learning irradiance forecasting models from sky images—An in-depth analysis. *Solar Energy* 224 (2021), 855–867.
- [35] Antonio Parretta, Angelo Sarno, and Luciano RM Vicari. 1998. Effects of solar irradiation conditions on the outdoor performance of photovoltaic modules. *Optics Communications* 153, 1-3 (1998), 153–163.
- [36] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2022. *Dataset shift in machine learning*. Mit Press.
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- [38] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricu, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).
- [39] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).
- [40] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfahl, Heather Cole-Lewis, Darlene Neal, et al. 2023. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617* (2023).
- [41] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [42] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [43] Yanxiang Wang, Jiawei Hu, Hong Jia, Wen Hu, Mahbub Hassan, Ashraf Uddin, Brano Kusy, and Moustafa Youssef. 2023. Spectral-Loc: Indoor Localization using Light Spectral Information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–26.
- [44] Zeyu Wang and Ravi S Srinivasan. 2017. A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. *Renewable and Sustainable Energy Reviews* 75 (2017), 796–808.
- [45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [46] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3 (2016), 1–40.
- [47] Yuxuan Weng, Guoquan Wu, Tianyue Zheng, Yanbing Yang, and Jun Luo. 2024. Large Model for Small Data: Foundation Model for Cross-Modal RF Human Activity Recognition. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. 436–449.
- [48] Matthew Wright and Ashraf Uddin. 2012. Organic–inorganic hybrid solar cells: A comparative review. *Solar energy materials and solar cells* 107 (2012), 87–111.
- [49] Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. 2024. NetLLM: Adapting Large Language Models for Networking. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 661–678.
- [50] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
- [51] Xuhai Xu, Bingsheng Yao, Yuanzhe Dong, Saadia Gabriel, Hong Yu, James Hendler, Marzyeh Ghassemi, Anind K Dey, and Dakuo Wang. 2024. Mental-llm: Leveraging large language models for mental health prediction via online text data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 1 (2024), 1–32.
- [52] Huanqi Yang, Sijie Ji, Rucheng Wu, and Weitao Xu. 2024. Are You Being Tracked? Discover the Power of Zero-Shot Trajectory Tracing with LLMs! *arXiv preprint arXiv:2403.06201* (2024).
- [53] Jonathan K Yohanna, Isaac N Itodo, and Victor I Umogbai. 2011. A model for determining the global solar radiation for Makurdi, Nigeria. *Renewable Energy* 36, 7 (2011), 1989–1992.
- [54] Jinsong Zhang, Rodrigo Verschae, Shohei Nobuhara, and Jean-François Lalonde. 2018. Deep photovoltaic nowcasting. *Solar Energy* 176 (2018), 267–276.
- [55] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 13001–13008.
- [56] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine* 5, 4 (2017), 8–36.

LightLLM: A Versatile Large Language Model for Predictive Light Sensing

- [57] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.