# Large Language Model Enabled Multi-Task Physical Layer Network

Tianyue Zheng, *Graduate Student Member, IEEE*, and Linglong Dai, *Fellow, IEEE*

***Abstract*—The advance of Artificial Intelligence (AI) is continuously reshaping the future 6G wireless communications. Particularly, the development of Large Language Models (LLMs) offers a promising approach to effectively improve the performance and generalization of AI in different physical-layer (PHY) tasks. However, most existing works finetune dedicated LLM networks for a single wireless communication task separately. Thus, performing diverse PHY tasks requires extremely high training resources, memory usage, and deployment costs. To solve the problem, we propose a LLM-enabled multi-task PHY network to unify multiple tasks with a single LLM, by exploiting the excellent semantic understanding and generation capabilities of LLMs. Specifically, we first propose a multi-task LLM framework, which finetunes LLM to perform multiple tasks including multi-user precoding, signal detection, and channel prediction. Besides, the multi-task instruction module, input encoders, as well as output decoders, are elaborately designed to distinguish different tasks and adapt LLM for different tasks in the wireless domain. Moreover, low-rank adaptation (LoRA) is utilized for LLM fine-tuning. To reduce the memory requirement during LLM fine-tuning, a LoRA fine-tuning-aware quantization method is introduced. Extensive numerical simulations are also displayed to verify the effectiveness of the proposed method.**

***Index Terms*—large language models (LLMs), multi-task LLM, physical layer communications**

## I. Introduction

The deep integration of Artificial Intelligence (AI) with wireless communications is one of the key features of Sixth-Generation (6G) communications [1]. With the development of 6G communications, the application of AI becomes essential to manage the exponentially growing data service. Attributed to the strong feature extraction capability, AI, especially deep learning (DL), has demonstrated great potential in a wide range of physical-layer (PHY) communication tasks [2], including channel state information (CSI) feedback [3], channel estimation [4], channel prediction [5], signal detection [6], etc. The AI Radio Access Network (AI-RAN) is anticipated to offer reduced latency, improved bandwidth, spectrum efficiency, and coverage. Moreover, significant efforts have been made by the 3rd Generation Partnership Project (3GPP) to standardize DL in wireless networks recently [7].

Despite the significant progress, existing DL empowered methods still face some fundamental issues that limit their applications in practical communication networks. First, with the dramatically increasing channel dimension and rapidly changing wireless environment in 6G applications, it is difficult for existing DL methods to comprehensively recognize patterns from complicated data distribution, due to their small model size and simple network structure. This insufficiency restricts their ability to provide reliable solutions in dynamic real-world scenarios. Secondly, existing DL methods exhibit a poor generalization capability to different wireless environments. For instance, a channel prediction model trained in an indoor scenario usually requires retraining when the wireless environment is changed to urban. Fortunately, recent advancements in large language models (LLMs) [8]–[10] have provided a radical solution to the challenges of existing DL methods. Bearing a huge amount of parameters, LLMs possess the ability to capture universal knowledge, which has demonstrated impressive language understanding and generation capabilities for various tasks in different domains.

Very recently, several initial studies [11]–[14] have been conducted since 2024 to leverage LLMs for boosting the performance of PHY communication systems. To be specific, in [11] LLM is adopted to perform power allocation using a few-shot learning approach. In this method, the channel gains and the corresponding transmit power strategies are provided as the input of LLM, also called "prompt" in the society of AI. The paper [11] demonstrates that LLM can automatically understand the principle of water-filling based optimal power allocation without any retraining. The authors in [12] effectively utilize LLMs to enhance AI-based CSI feedback performance in various scenarios. They incorporate the channel distribution as a prompt within the decoder to further enhance channel reconstruction quality. Besides, in order to obtain more accurate channel prediction and improve the generalization capability, the authors in [13] propose an LLM-driven channel prediction approach. [14] also unleashes the strength of LLMs for time series forecasting to improve the robustness of beam prediction. Moreover, a few review papers [15]–[19] have explored the potential transformative impact of LLMs and provide envisions for LLM-aided wireless communications.

However, most existing works finetune dedicated LLM networks for a *single* wireless communication task, such as the previously mentioned channel prediction and CSI feedback, etc. In reality, wireless communication systems involve a multitude of tasks, each with distinct requirements that necessitate the adoption of LLMs. Given the huge model size of LLM, designing and training dedicated models for each task separately will lead to extremely high computational

This work was supported in part by the National Science Fund for Distinguished Young Scholars (Grant No. 62325106), in part by the Key Program of the National Natural Science Foundation of China (Grant No. 62031019), and in part by the National Key Research and Development Program of China (Grant No. 2023YFB3811503).

T. Zheng, and L. Dai are with the Department of Electronic Engineering, Tsinghua University, and the State Key laboratory of Space Network and Communications, Tsinghua University, Beijing 100084, China (e-mails: zhengty22@mails.tsinghua.edu.cn, daill@tsinghua.edu.cn).

complexity, memory usage, and deployment costs.

To address the problem, we propose a LLM-enabled multi-task PHY network which retains the advantages of LLMs compared to small models while reducing training and deployment costs. Leveraging the excellent language understanding and generation capabilities of LLMs, the framework *unifies* multiple tasks with a *single* LLM. The main contributions of this paper are summarized as follows[1].

- We propose a multi-task LLM framework for PHY communications. The framework enables us to input different task requirements to LLMs with natural language, and explore the global feature extraction ability of LLMs to execute multiple tasks within one network. Particularly, this work focuses on three of the typical tasks in PHY communications: multi-user precoding, signal detection, and channel prediction. The proposed framework can also accommodate any other PHY tasks.
- For the design of the proposed framework, dedicated modules are proposed to adapt the LLMs for multiple tasks in the wireless domain. It mainly consists of two components: multi-task instructions and task-specific encoders and decoders. Specifically, first, in order to distinguish and cope with different tasks and different data formats of the tasks, we elaborately design multi-task instructions as prompts of LLM; secondly, to bridge the gap between the features of wireless data and those of the LLM, task-specific encoders and decoders are proposed to adapt the text-based pre-trained LLM.
- For the fine-tuning of the proposed framework, we introduce low-rank adaptation (LoRA) [20], which freezes the pre-trained model weights and injects trainable low-rank matrices. Furthermore, to mitigate the computational and memory demands of the proposed model, we employ a LoRA fine-tuning-aware quantization method [21] that simultaneously quantizes an LLM and finds a proper low-rank initialization for LoRA fine-tuning.
- Extensive simulation experiments have been conducted to verify the effectiveness of the proposed method. Our proposed multi-task LLM outperforms the majority of task-specific baselines across all tasks. Compared to dedicated LLMs designed for each task, our proposed multi-task LLM achieves comparable performance. In addition, the LoRA and LoRA fine-tuning-aware quantization method achieve significant resource reduction regarding the trainable parameters, and memory usage. Lastly, the proposed multi-task LLM exhibits superior few-shot learning and generalization ability.

The rest of the paper is organized as follows. Section II introduces the system model. Then, the three selected tasks, namely multi-user precoding, signal detection, and channel prediction, are formulated respectively. In Section III, the design of the proposed multi-task PHY LLM is illustrated in detail. Besides, Section IV elaborates on the fine-tuning strategy of the proposed multi-task PHY LLM. Simulation results are provided in Section V. Finally, Section VI concludes this paper.

[1]Simulation codes will be provided to reproduce the results in this paper: http://oa.ee.tsinghua.edu.cn/dailinglong/publications/publications.html.

*Notation:* $\mathbf{a}^H$, $\mathbf{A}^H$ denote the conjugate transpose of vector $\mathbf{a}$ and matrix $\mathbf{A}$, respectively; $\|\mathbf{a}\|_2$ denotes the $l_2$ norm of vector $\mathbf{a}$; $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix $\mathbf{A}$; $\mathbb{R}$,$\mathbb{C}$ denote the set of real numbers and complex numbers, respectively; $\mathcal{CN}(\mu, \mathbf{\Sigma})$ denotes the probability density function of complex multivariate Gaussian distribution with mean $\mu$ and variance $\mathbf{\Sigma}$.

## II. Systems Model

We consider a multi-user (MU) multiple-input-single-output (MISO)- orthogonal-frequency-division-multiplexing (OFDM) system working in a time-division-duplex (TDD) mode. A base station (BS) simultaneously serves $K$ single-antenna mobile users. The BS is equipped with a uniform planar array (UPA) consisting of $N_{\mathrm{T}} = N_h \times N_v$ antennas, where $N_h$ and $N_v$ denote the number of antennas along the horizontal and vertical dimensions, respectively. To design a multi-task LLM for the BS, we select typical tasks in PHY communications for illustration based on the following rationales. First, to fully demonstrate the multitasking capability of LLM, we aim to select distinct tasks with varying objectives, data. Secondly, we attempt to select tasks that are essential within the transceivers. Nevertheless, these tasks should be challenging, necessary, and suitable for Transformer-based LLMs to solve. Based on these rationals, we choose downlink multi-user precoding, uplink signal detection, and channel prediction for mobile devices, respectively, in this work. The system models and problem descriptions of these three tasks are presented below.

### A. Multi-user Precoding

For downlink transmission scenario, the channel between user $k$ and the BS at the $m$-th subcarrier is denoted as $\mathbf{h}_k^m \in \mathbb{C}^{N_t \times 1}, m = 1, 2, \cdots, M$. The received signal of user $k$ at the $m$-th subcarrier is given by

$$y_k^m = \mathbf{h}_k^{mH} \sum_{k'=1}^{K} \mathbf{w}_{k'}^m x_{k'}^m + n_k^m, \tag{1}$$

where $\mathbf{w}_k^m$ represents the beamforming vector for user $k$, $x_k^m$ with $\mathbb{E}(|x_k^m|^2) = 1$, is the transmitted symbol from the BS to user $k$, and $n_k^m \sim \mathcal{CN}(0, \sigma^2)$ denotes the additive Gaussian white noise (AWGN) with zero mean and variance $\sigma^2$.

Multi-user precoding aims to maximize the system sum rate via the optimization of the transmit precoders. The total power of all beamforming vectors is limited due to the BS power budget. For simplicity, we design the beamformers based on the channel of the central carrier-frequency $\mathbf{h}_k$, and the problem is mathematically formulated as

$$\max_{\mathbf{W}} \sum_{k=1}^{K} \log_2(1 + \gamma_k), \quad \text{s.t.} \sum_{k=1}^{K} \|\mathbf{w}_k\|^2 \leq P_{\max}, \tag{2}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K]$ is a set of beamforming vectors and $P_{\max}$ is the power budget. Besides, $\gamma_k$ represents

the received signal-to-interference-plus-noise ratio (SINR) at user $k$. It is written as

$$\gamma_k = \frac{\left|\mathbf{h}_k^H \mathbf{w}_k\right|^2}{\sum_{k'=1,k'\neq k}^{K} \left|\mathbf{h}_k^H \mathbf{w}_{k'}\right|^2 + \sigma^2}. \tag{3}$$

As pointed out in [22], the optimal downlink beamforming vectors for (2) follow the structure as

$$\mathbf{w}_k^* = \sqrt{p_k} \frac{\left(\mathbf{I}_{N_\mathrm{T}} + \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2} \mathbf{h}_k \mathbf{h}_k^H\right)^{-1} \mathbf{h}_k}{\left\| \left(\mathbf{I}_{N_\mathrm{T}} + \sum_{k=1}^{K} \frac{\lambda_k}{\sigma^2} \mathbf{h}_k \mathbf{h}_k^H\right)^{-1} \mathbf{h}_k \right\|_2}, \quad \forall k, \tag{4}$$

where $p_k$ is the power allocated to the $k$-th use, and $\lambda_k$ is a positive parameter and $\sum_{k=1}^{K} \lambda_k = \sum_{k=1}^{K} p_k = P_{\max}$. The solution structure in (4) provides the required expert knowledge for the LLM-empowered beamforming design in (2). In conventional approaches, the WMMSE algorithm has been widely adopted as an effective solution. However, two fundamental limitations persist in the WMMSE algorithm. First, WMMSE inherently converges to a local optimal solution, resulting in a performance gap compared to the globally optimal solution. Secondly, the iterative nature of WMMSE and the inversion of a high-dimensional matrix in each iteration introduce prohibitive execution delays in real-time deployments. To address these challenges, we introduce LLM for multi-user precoding. Given this knowledge in (4), the LLM is only required to learn $2K$ key parameters $\lambda = [\lambda_1, \lambda_2, \cdots, \lambda_K]$ and $\mathbf{p} = [p_1, p_2, \cdots, p_K]$, instead of the whole $K \times N$ beamforming matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K]$. Thus, the problem of DL-based multi-user precoding can be reformulated as

$$\textbf{P1:} \min_{\Omega_{\mathrm{PRE}}} \quad \sum_{k=1}^{K} \log_2(1+\gamma_k), \tag{5}$$

$$\text{s.t.} \quad \mathbf{w}_k \text{ in (4)}, \ \hat{\lambda}, \hat{\mathbf{p}} = f_{\Omega_{\mathrm{PRE}}}(\mathbf{H}), \tag{6}$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_K]$. Here, we suppose accurate multi-user channel $\mathbf{H}$ can be acquired. $\hat{\mathbf{p}}, \hat{\lambda}$ is learned from the neural network with mapping function $f_{\Omega_{\mathrm{PRE}}}$, where $\Omega_{\mathrm{PRE}}$ is the learnable parameters.

### B. Signal Detection

During uplink transmission, the BS antenna array simultaneously receives the transmitted symbols from the $K$ users. For clarity, we introduce the superscript "$\bar{\cdot}$" to all parameters related to the uplink transmission, e.g., $\bar{a}$. Specifically, we denote the transmitted symbol vector from all users at the $m$-th subcarrier as $\bar{\mathbf{x}}^m = [\bar{x}_1^m, \bar{x}_2^m, \cdots, \bar{x}_K^m] \in \mathbb{C}^{K\times 1}$. Each element is drawn from the $P$-QAM constellation, and then transmitted over the channel. The received signal $\bar{\mathbf{y}}^m \in \mathbb{C}^{N_\mathrm{T}\times 1}$ is

$$\bar{\mathbf{y}}^m = \bar{\mathbf{H}}^m \bar{\mathbf{x}}^m + \bar{\mathbf{n}}^m, \tag{7}$$

where $\bar{\mathbf{H}}^m = [\bar{\mathbf{h}}_1^m, \bar{\mathbf{h}}_2^m . \cdots, \bar{\mathbf{h}}_K^m]$ is the uplink channel of the $K$ users and $\bar{\mathbf{n}}^m \sim \mathcal{CN}(0, \bar{\sigma}^2 \mathbf{I}_{N_\mathrm{T}})$ is the AWGN at the $m$-th subcarrier.

BS requires to recover the signals $\bar{\mathbf{x}}^m$ from the received signal $\bar{\mathbf{y}}^m$ given $\bar{\mathbf{H}}^m$. We adopt the minimum mean squared error (MMSE) estimator to formulate the associated signal detection problem as

$$\textbf{P2:} \min_{\Omega_{\mathrm{DET}}} \quad ||\hat{\bar{\mathbf{x}}}^m - \bar{\mathbf{x}}^m||_2 \tag{8}$$

$$\text{s.t.} \quad \hat{\bar{\mathbf{x}}}^m = f_{\Omega_{\mathrm{DET}}}(\bar{\mathbf{H}}^m, \bar{\mathbf{y}}^m), \tag{9}$$

where $f_{\Omega_{\mathrm{DET}}}$ is the mapping function with variable parameters $\Omega_{\mathrm{DET}}$.

### C. Channel Prediction

In mobile scenarios involving high-velocity users, it is possible that the channel coherence time is shorter than the channel estimation period. Under these circumstances, precise channel prediction becomes crucial to mitigate the channel aging phenomenon [23] in high-mobility communication environments with rapidly changing channels. In this work, we aim to accurately predict the CSI of the next $T_2$ time slots given the CSI of the previous $T_1$ time slots. The CSI of $M$ subcarriers at time $t$ is represented in matrix form as

$$\mathbf{H}_k^t = [\mathbf{h}_k^{1,t}, \mathbf{h}_k^{2,t}, \cdots, \mathbf{h}_k^{M,t}], \forall k, t, \tag{10}$$

where $\mathbf{h}_k^{m,t}$ is the channel of user $k$, time $t$, and subcarrier $m$.

To evaluate the channel prediction accuracy, the normalized MSE (NMSE) between predicted CSI by the network and ground-truth CSI is selected as the metric. Utilizing the metric, the channel prediction problem can be described as follows:

$$\textbf{P3:} \min_{\Omega_{\mathrm{CP}}} \quad \frac{\sum_{t=1}^{T_2} \|\hat{\mathbf{H}}^{t_0+t} - \mathbf{H}^{t_0+t}\|_F^2}{\sum_{t=1}^{T_2} \|\mathbf{H}^{t_0+t}\|_F^2} \tag{11}$$

$$\text{s.t.} \quad (\hat{\mathbf{H}}^{t_0+1}, ..., \hat{\mathbf{H}}^{t_0+T_2}) = f_{\Omega_{\mathrm{CP}}}(\mathbf{H}^{t_0}, ..., \mathbf{H}^{t_0-T_1+1}), \tag{12}$$

where $\hat{\mathbf{H}}^t$ is the predicted CSI, and $f_{\Omega_{\mathrm{CP}}}$ is the mapping function of the network with trainable parameters $\Omega_{\mathrm{CP}}$. Note here we ignore the subscript since the channel prediction is performed independently to each user.

## III. The Framework of the Proposed Physical Layer Multi-task LLM

In this paper, we propose an LLM-enabled PHY network to *unify multiple tasks* (channel prediction, multi-user precoding, and signal detection for instance) with a *single* network. The proposed framework is illustrated in Fig. 1, including a multi-task instruction module, an input encoder, an LLM backbone, and an output decoder. To distinguish and cope with different tasks, we design multi-task instructions as prompts of LLM, which are processed by the pre-trained LLM embedder as part of the LLM input. Besides, the wireless data is encoded by task-specific encoders to make the feature of *wireless data* understandable to a *text-based* pre-trained LLM, e.g., GPT-2. The prompt and encoded wireless data are concatenated together to serve as the inputs of the LLM. These inputs are later fed into the LLM backbone. Note that the same LLM backbone is used for all considered tasks. Finally, obtaining the outputs of the LLM backbone, task-specific decoders are utilized to generate desired outputs for different tasks.

We describe each component of the framework as follows. It is worth noting that, in this paper, we select three typical

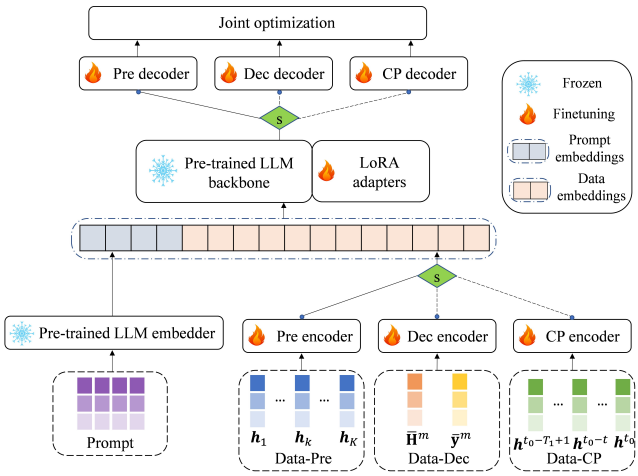


Fig. 1. The model framework of our method. In this figure, "CP" denotes channel prediction, "Det" denotes signal detection, and "Pre" denotes multi-user precoding.

tasks suitable for DL methods in PHY communications, while the proposed method can be smoothly extended to other PHY tasks. Besides, we mention that although the encoders and decoders are designed specifically for each task, the parameter sizes of these modules are much smaller than the LLM backbone. The LLM backbone, which takes up the majority of the total parameter size, is shared among different tasks.

### A. Multi-task Instruction Template

When training a single unified model for multiple different tasks, a basic and critical problem is how to distinguish each task. Thanks to the remarkable text understanding capability, LLM allows users to input task requirements with natural language. Therefore, we can design a multi-task instruction template with task-specific tokens as prompts to make each task easily distinguishable. Next, we introduce our multi-task instruction template in detail.

We structure the instruction template into three parts, which are denoted as follows,

$$[Task\ Identifier]\text{Task description} < Instruction > . \tag{13}$$

The first part is the task identifier token, the second part is the task and data description, and the third part is the instruction input. **Task Identifier** provides a distinct identifier for each task to reduce the ambiguity across various tasks. Based on the task identifier, the model can distinguish different tasks and activate the corresponding modules of the encoder and decoder. **Task and Data Description** attempts to input basic wireless domain knowledge to the model. It improves the LLM's comprehension of the targeted task and is promising to accelerate convergence during training. Take multi-user precoding as an example, the designed task description is presented by "*For the collected dataset, we consider a BS with 128 antennas to serve 8 single antenna users simultaneously*". The third part **Instruction** gives a direct and clear objective of the task. Again, for instance, the instruction for multi-user precoding is "*<Instruction> Design the precoding matrix given channels of the users, to maximize the sum rate of the multiple users.*".

The designed prompt is then fed to the pre-trained LLM embedder as part of the input of the LLM. The prompt embedding is denoted as $\mathbf{X}^{\text{emb}}_{\text{prompt},n}$ for task $n$, where $n$ is the task index, denoting one of CP, Det and Pre.

### B. Input Encoders

Adapting text-based pre-trained LLM to multiple communication domain tasks is another challenging problem. To be specific, first, there is a huge gap between the specific characteristics of the data in the communication domain and the natural language. Therefore, directly applying a general-domain LLM to these communication tasks may lead to poor performance. Secondly, the formats, distributions, and feature spaces of the data also differ significantly among the tasks. Therefore, task-specific encoders are required to perform task alignment operations, enabling the same LLM backbone to handle different tasks.

In this subsection, we elaborate on the designed input encoder modules. To facilitate multi-task feature extraction, the design of task-specific encoders is mainly based on two principles: task-customized and lightweight. To be specific, the encoder architecture should be customized according to each task's data characteristics and objectives to optimize feature extraction capabilities for different tasks. Besides, given the task-specific nature of these modules, we attempt to design the encoders as lightweight as possible to avoid high computational overheads. Next, we elaborate on the design of input encoders in detail.

*1) Encoder for multi-user precoding:* Since neural networks generally deal with real numbers, we first convert the complex multi-user channel into real tensors $\mathbf{X}^{\text{in}}_{\text{Pre}} \in \mathbb{R}^{K\times 2N_t}$ as input. For multi-user precoding, the encoder should perform joint feature extraction from multi-user CSI while capturing inter-user channel correlations. The Transformer architecture is particularly suitable for this objective due to its inherent strength in modeling relational patterns through self-attention mechanisms. Thus, we employ a shallow Transformer encoder composed of $L = 3$ blocks. The structure of each block is presented in Fig. 2, including a multi-head self-attention module and a multilayer perceptron (MLP) module. Input $\mathbf{X}^{\text{in}}_{\text{Pre}}$ is first processed by the multi-head self-attention module:

$$\mathbf{X}^{\text{att}(1)}_{\text{Pre}} = \text{LayerNorm}(\text{ATT}(\mathbf{X}^{\text{in}}_{\text{Pre}}) + \mathbf{X}^{\text{in}}_{\text{Pre}}), \tag{14}$$

where $\text{ATT}(\cdot)$ denotes the multi-head self-attention, $\text{LayerNorm}(\cdot)$ denotes the layer normalization across the feature dimension, applied after residual connections. Then the MLP module is performed as

$$\mathbf{X}^{\text{mlp}(1)}_{\text{Pre}} = \text{LayerNorm}(\text{MLP}(\mathbf{X}^{\text{att}(1)}_{\text{Pre}}) + \mathbf{X}^{\text{att}(1)}_{\text{Pre}}), \tag{15}$$

which serves as the input of the next block. $\text{MLP}(\cdot)$ denotes the MLP processing. Denote the output of the transfomer as $\mathbf{X}^{\text{mlp}(3)}_{\text{Pre}}$. Then a linear layer is used to project $\mathbf{X}^{\text{mlp}(3)}_{\text{Pre}}$ into the language model space

$$\mathbf{X}^{\text{emb}}_{\text{Pre}} = \text{Linear}(\mathbf{X}^{\text{mlp}(3)}_{\text{Pre}}). \tag{16}$$

*2) Encoder for signal detection:* The input of the signal detection encoder contains the uplink channel $\bar{\mathbf{H}}^m$ and the

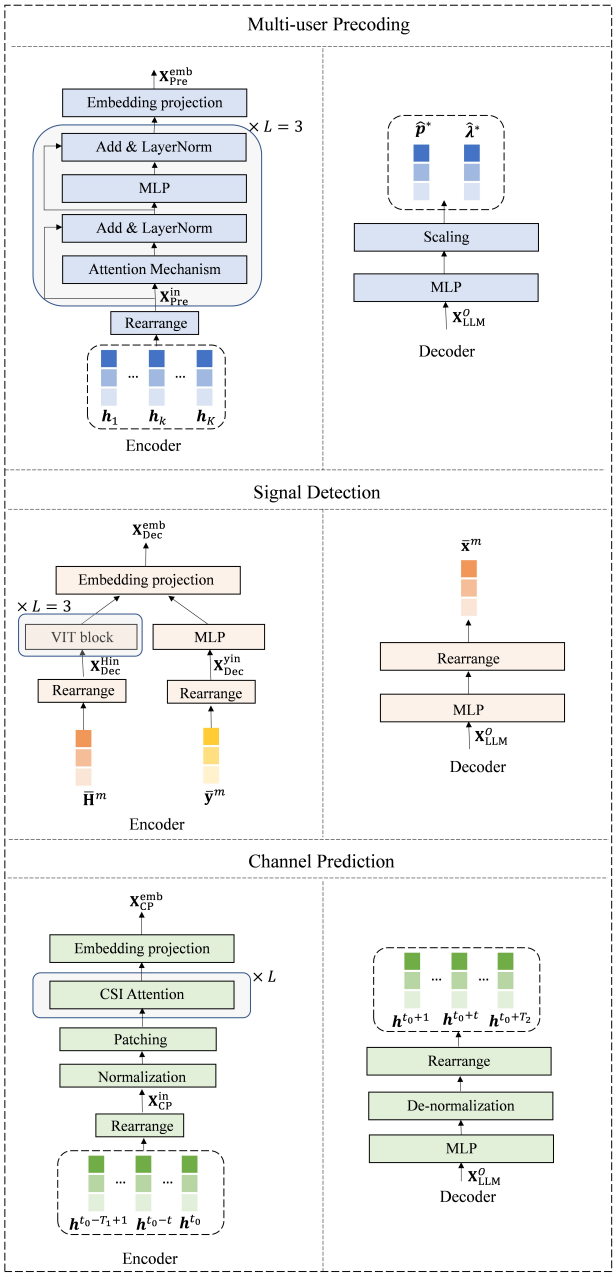


Fig. 2. Detailed illustration of encoders and decoders.

received signal $\bar{\mathbf{y}}^m$. Given their distinct statistical distributions, we thereby perform initial feature extraction for CSI and received symbols, respectively. Specifically, the encoder rearranges the channel $\bar{\mathbf{H}}^m$ as one token $\mathbf{X}_{\text{Det}}^{\text{Hin}} \in \mathbb{R}^{K\times 2N_t}$. Then, a shallow vision transformer-based encoder consisting of $L = 3$ transformer blocks is employed to extract features from the 2D channel matrix:

$$\mathbf{X}_{\text{Det}}^{\text{Hvit}} = \text{VIT}(\mathbf{X}_{\text{Det}}^{\text{Hin}}), \tag{17}$$

where $\text{VIT}(\cdot)$ denotes the vision transformer encoder [24]. We also extract shallow features from the rearranged real received signal $\mathbf{X}_{\text{Det}}^{\text{yin}}$ by an MLP module

$$\mathbf{X}_{\text{Det}}^{\text{ymlp}} = \text{MLP}(\mathbf{X}_{\text{Det}}^{\text{yin}}). \tag{18}$$

In the simulation parts, we find that integrating received signals of several time slots into one sample effectively improves the training performance compared to treating them as multiple samples. Therefore, we utilize received signals of $L_0 = 8$ time slots in one sample in practice. Then, the extracted features of the channel and received data are concatenated and projected to the input format of LLM:

$$\mathbf{X}_{\text{Det}}^{\text{emb}} = \text{Linear}([\mathbf{X}_{\text{Det}}^{\text{Hvit}}, \mathbf{X}_{\text{Det}}^{\text{ymlp}}]). \tag{19}$$

*3) Encoder for channel prediction:* Consider that the CSI $\mathbf{H}^t \in \mathbb{C}^{N_t\times M}$ of at time $t$ is the high-dimensional structural data, directly predicting the matrix by the network will bring significant complexity. The complexity can be unacceptable for future 6G systems with a large number of antennas and subcarriers. Inspired by [13], we parallelize the channel prediction for different antennas. That is to say, we predict the CSI of each transmitter antenna separately. Thereby, the input sample of $j$-th antenna, for $j \in \{1, 2, \cdots, N_T\}$, can be converted into a real tensor $\mathbf{X}_{\text{CP}}^{\text{in}} \in \mathbb{R}^{T_1\times 2M}$. To facilitate convergence of network training, we first perform batch normalization for the input data as $\mathbf{X}_{\text{CP}}^{\text{in}'}$. Then the normalized input $\mathbf{X}_{\text{CP}}^{\text{in}'}$ is divided into $T_1' = \lceil \frac{T_1}{N} \rceil$ non-overlapping patches of size $N$ along the temporal dimension [25]. The patching operation helps to capture the local temporal features, and the patched input is denoted as $\mathbf{X}_{\text{CP}}^{\text{pat}} \in \mathbb{R}^{T_1'\times N\times 2M}$. The input necessitates an encoder architecture employing operations that effectively capture high-dimensional features. Thus, we adopt the CSI attention module proposed in [26], to extract preliminary temporal and frequency features before LLM:

$$\mathbf{X}_{\text{CP}}^{\text{CA}} = \text{CSIATT}^L(\mathbf{X}_{\text{CP}}^{\text{pat}}), \tag{20}$$

where $\text{CSIATT}^L$ represents the CSI attention module cascaded $L$ times. Finally, to align with the input format of LLM backbone, $\mathbf{X}_{\text{CP}}^{\text{CA}}$ is rearranged to $\mathbf{X}_{\text{CP}}^{\text{CA}'} \in \mathbb{R}^{T_1'\times 2MN}$ and then mapped to the feature dimension of the pre-trained LLM with a single fully-connected layer:

$$\mathbf{X}_{\text{CP}}^{\text{emb}} = \text{Linear}(\mathbf{X}_{\text{CP}}^{\text{CA}'}). \tag{21}$$

## C. LLM Mainbody

In this paper, we adopt GPT-2 [8] as the LLM backbone for our proposed multi-task PHY framework. It should be emphasized that our framework allows for seamless integration of alternative LLMs, including but not limited to LLAMA [9] and QWen [27]. The decision of model architecture and scale requires evaluation of the trade-off between computational complexity and performance.

The architecture of GPT-2 consists of stacked classical transformer decoders, which is presented in Fig. 3. Note that in the proposed multi-task PHY network, the LLM backbone takes up most of the model parameters. And the backbone and its parameters remain shared across all tasks. In contrast, the encoders and decoders, though designed specifically for each task, only occupy a very small portion of the network. To be specific, the parameter size of the GPT-2 backbone is 124 million, while the parameter size of all other modules, is only about 18 million. Obtaining the LLM backbone, the embeddings of data of task $n$ and the corresponding prompt

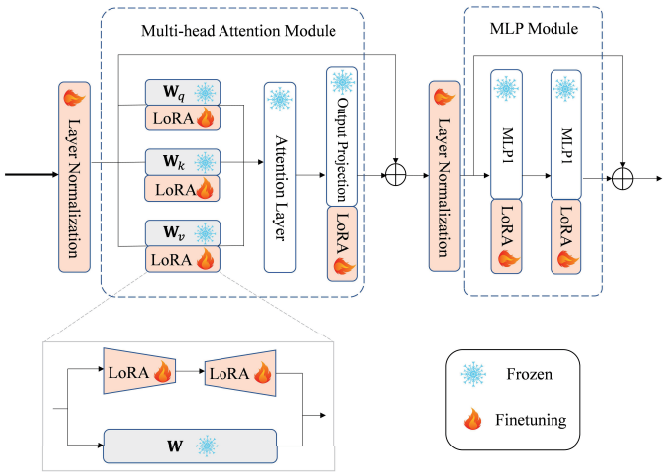


Fig. 3. The illustration of a transformer block in GPT-2.

are concatenated as $\mathbf{X}_n^{\mathrm{I}} = [\mathbf{X}_{\mathrm{prompt},n}^{\mathrm{emb}}, \mathbf{X}_n^{\mathrm{emb}}]$]. Then the input of each task is individually fed to the shared LLM backbone. The output of task $n$ can be obtained by

$$\mathbf{X}_n^{\mathrm{LLM}} = \mathrm{LLM}(\mathbf{X}_n^{\mathrm{I}}). \tag{22}$$

The outputs of three tasks are processed by output decoders and then utilized for joint optimization.

### D. Output Decoders

Similar to input encoders, to facilitate multiple downstream tasks simultaneously, task-specific decoders are required to convert the output features of the LLM into the final results for different tasks. Here, we elaborate on the output decoders of the three tasks sequentially. Note that for output decoders, we discard the prefix portion that is associated with the instruction prompt, while we only retain the output presentations of the data for the decoder.

*1) Decoder for multi-user precoding:* As discussed in Section II A., the multi-user precoding problem can be transformed into the learning of the parameters $\lambda = [\lambda_1, \lambda_2, \cdots, \lambda_K]$ and $\mathbf{p} = [p_1, p_2, \cdots, p_K]$. Therefore, an MLP module comprising two fully-connected (FC) layers is adopted to generate $2K$ values, including the power allocation vectors $\hat{\lambda}$ and $\hat{\mathbf{p}}$. The output of the MLP module $\mathbf{X}_{\mathrm{Pre}}^O \in \mathbb{R}^{K\times 2}$ can be written as

$$\mathbf{X}_{\mathrm{Pre}}^O = \mathrm{MLP}(\mathbf{X}_{\mathrm{Pre}}^{\mathrm{LLM}}), \tag{23}$$

where $\hat{\lambda} = \mathbf{X}_{\mathrm{Pre}}^O[:,1]$ and $\hat{\mathbf{p}} = \mathbf{X}_{\mathrm{Pre}}^O[:,2]$. Then the scaling layer scales the results of the output layer $\hat{\lambda}$ and $\hat{\mathbf{p}}$ to meet the power constraint by:

$$\hat{\mathbf{p}}^* = \frac{P_{\max}}{\|\hat{\mathbf{p}}\|_1}\hat{\mathbf{p}} \quad \text{and} \quad \hat{\lambda}^* = \frac{P_{\max}}{\|\hat{\lambda}\|_1}\hat{\lambda}. \tag{24}$$

*2) Decoder for signal detection:* For signal detection, we employ two FC layers to transform the dimension of the $\mathbf{X}_{\mathrm{LLM}}^O$ to the number of antennas:

$$\mathbf{X}_{\mathrm{Det}}^{\mathrm{mlp}} = \mathrm{MLP}(\mathbf{X}_{\mathrm{Det}}^{\mathrm{LLM}}). \tag{25}$$

Then $\mathbf{X}_{\mathrm{Det}}^{\mathrm{mlp}}$ is rearranged $\mathbf{X}_{\mathrm{Det}}^O \in \mathbb{R}^{K\times 2}$ to where the first and the second dimension respectively correspond to the real part and the imaginary part.

*3) Decoder for channel prediction:* We utilize an MLP module as the decoder to predict the channel. The output is presented by

$$\mathbf{X}_{\mathrm{CP}}^{\mathrm{mlp}} = \mathrm{MLP}(\mathbf{X}_{\mathrm{CP}}^{\mathrm{LLM}}). \tag{26}$$

Last, $\mathbf{X}_{\mathrm{CP}}^{\mathrm{mlp}}$ is de-normalized to generate the final output of the network, i.e.,

$$\mathbf{X}_{\mathrm{CP}}^{norm} = \sigma_{\mathrm{CP}}\mathbf{X}_{\mathrm{CP}}^{\mathrm{mlp}} + \mu_{\mathrm{CP}}, \tag{27}$$

where $\mu_{\mathrm{CP}}$ and $\sigma_{\mathrm{CP}}$ is the mean and variance of the channel matrix. Then the tensors are rearranged to $\mathbf{X}_{\mathrm{CP}}^O \in \mathbb{R}^{T_2\times M\times 2}$ to separate the real and imaginary part of the predicted channel.

## IV. The Fine-tuning Strategy of the Proposed Physical Layer Multi-task LLM

Considering the huge parameter size of LLMs, directly fine-tuning all the parameters of the proposed network is impractical. To address this issue, this section focuses on the computationally efficient fine-tuning of the proposed PHY multi-task LLM. First, we introduce LoRA [20] for LLM fine-tuning, which inserts *low-rank adapters* into the LLM backbone to finetune the pre-trained LLM for PHY tasks. Then, to mitigate the computational and memory demands of the proposed model, we employ a LoRA fine-tuning-aware *quantization method* [21] that performs LLM quantization while concurrently properly initializing the low-rank adapters for LoRA-based fine-tuning. Moreover, the multi-task loss as well as the training schedule are illustrated.

### A. LoRA for LLM Mainbody

Existing strategies of using LLMs in the wireless communication domain can be divided into two categories. The first is direct application of pretrained LLM [11] or only LayerNorm fine-tuning [13], which limits the adaptability to wireless tasks. The other strategy is full-parameter finetuning [3]. It enables LLMs specific to wireless communications, but introduces prohibitive cost in practice.

Therefore, in this work, we introduce an efficient LLM fine-tuning technique, LoRA [20], for the proposed multi-task LLM, enhancing tunable capacity while maintaining efficiency. To be specific, LoRA is a parameter-efficient technique that freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer. The use of LoRA mainly brings two advantages. Firstly, due to the immense parameter size of LLMs, full-parameter fine-tuning could be impractical. LoRA circumvents this by only fine-tuning low-rank weight matrices, with a significantly reduced number of parameters. Secondly, LoRA prevents the problem of catastrophic forgetting of the original knowledge during fine-tuning. This is attributed to the fact that the newly learned knowledge has a lower rank than the original knowledge. As a result, LoRA facilitates the use of the universal modeling and generalization capability of pre-trained LLMs to achieve multiple tasks with a single model.

Specifically, we focus on the fine-tuning of parameters of linear projection layers in both multi-head attention module and MLP module of the transformer block, including query matrices $\mathbf{W}_q$, key matrices $\mathbf{W}_k$, value matrices $\mathbf{W}_v$, output projection matrices in multi-head attention module $\mathbf{W}_o$ and two linear projection matrices in MLP module $\mathbf{W}_{up}$, $\mathbf{W}_{down}$, as illustrated in Fig. 3. LoRA updates two rank decomposition weight matrices $\mathbf{A}$ and $\mathbf{B}$ that are attached to a frozen pre-trained weight matrix $\mathbf{W}$. In this case, the original weight $\mathbf{W}$ is modified as

$$\mathbf{W} \leftarrow \mathbf{W} + \mathbf{A}\mathbf{B}^T, \tag{28}$$

where $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$, $\mathbf{B} \in \mathbb{R}^{d_2 \times r}$, and $r \ll \min\{d_1, d_2\}$. Generally, we initialize the weights as follows:

$$\mathbf{A} \sim \mathcal{N}(0, \sigma^2), \quad \mathbf{B} = 0. \tag{29}$$

As mentioned above, during the fine-tuning, we freeze $\mathbf{W}$ while updating $\mathbf{A}$ and $\mathbf{B}$. Since $r << \min(d_1, d_2)$, the number of parameters for fine-tuning $\mathbf{A}$ and $\mathbf{B}$, i.e., $rd_1 + rd_2$, is significantly less than that of the full weight matrix, $d_1 d_2$. The reduced parameter size thus makes the LoRA-based fine-tuning much efficient.

Besides, according to [28], learning task-specific Layer Normalization can significantly improve the performance for various tasks while only adding a few parameters, so we also finetune the parameters of normalization layers in all transformer blocks.

### *B. LoRA Fine-tuning-aware Quantization*

Despite the usage of LoRA, extensive computational and memory demands of LLM-based models still pose significant challenges for both fine-tuning and inference, especially for resource-restricted equipment in wireless communications. To reduce the storage demands of pre-trained models, quantization acts as an essential compression strategy. Its key idea is to transform the original weights with high precision into a finite set of discrete values. For instance, converting model parameters from the original 16-bit floating-point format (FP16) to a 4-bit integer format leads to a 75% decrease in storage overhead. However, it is worth noting that quantized network parameters might degrade the performance of the aforementioned LoRA fine-tuning strategy. Particularly, consider a quantized weight matrix $\mathbf{Q} = q(\mathbf{W})$, where $q(\cdot)$ denotes the quantization operator. If the low-rank adapters $\mathbf{A}$ and $\mathbf{B}$ are initialized by (29) and then attached to $\mathbf{Q}$, the initial weight matrix $\mathbf{Q} + \mathbf{A}\mathbf{B}^T$ naturally diverges from the original pre-trained parameters $\mathbf{W}$ because of the variations caused by the quantization process. Such unavoidable variations may negatively influence the initial setup of the LoRA fine-tuning procedure. What's more, wireless tasks demand rigorous numerical precision than text generation. Thus, the influence of quantization in wireless domains remains unverified.

To solve the problem, we adopt a LoRA-fine-tuning-aware method to smoothly integrate quantization into the procedure of LoRA fine-tuning. The main idea is to approximate the original high-precision pre-trained weights by alternatively applying quantization for LLMs and proper low-rank initialization for LoRA fine-tuning. This initialization strategy effectively reduces the gap between the quantized and full-precision model, leading to enhancement in fine-tuning performance.

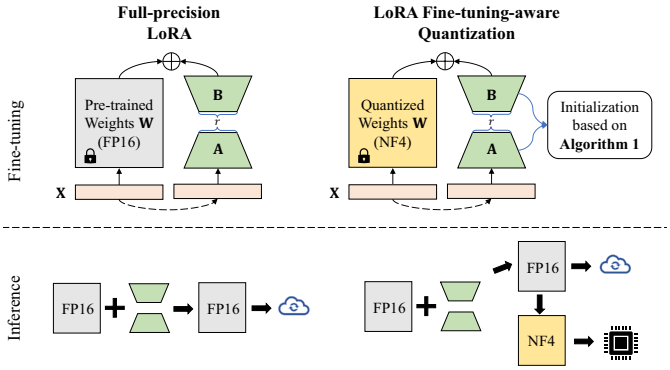


Fig. 4. The comparison of LoRA fine-tuning-aware quantization with traditional full-precision LoRA.

The key procedures of the LoRA fine-tuning-aware quantization method are presented in Fig. 4. During the initialization of LoRA fine-tuning, a quantized weight matrix $\mathbf{Q} \in \mathbb{R}^{d_1 \times d_2}$ with $N$-bit precision, along with the low-rank adapters $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$, $\mathbf{B} \in \mathbb{R}^{d_2 \times r}$ are designed to closely approximate the original full-precision pre-trained parameter matrix $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$. Mathematically, the model weight initialization problem can be formulated as:

$$\textbf{P4:} \min_{\mathbf{Q},\mathbf{A},\mathbf{B}} \|\mathbf{W} - \mathbf{Q} - \mathbf{A}\mathbf{B}^T\|_F. \tag{30}$$

This problem can be efficiently solved by alternatively conducting quantization and singular-value decomposition. The step-by-step procedures provided in **Algorithm 1** and the details are as follows.

**Quantization Step:** In the $i$-th iteration, the quantization process is applied to the residual between the pre-trained parameter matrix $\mathbf{W}$ and the low-rank approximation $\mathbf{A}_{i-1}\mathbf{B}_{i-1}^T$, yielding the quantized weight matrix $\mathbf{Q}_i$

$$\mathbf{Q}_i = q_N(\mathbf{W} - \mathbf{A}_{i-1}\mathbf{B}_{i-1}^T), \tag{31}$$

where $q_N(\cdot) : \mathbb{R} \mapsto \{0, 1, \cdots, 2^N - 1\}$ maps a high-precision weight matrix, e.g., a matrix with 16-bit floating point numbers, to an $N$-bit quantized matrix. Typically, the quantization process can be expressed as

$$\mathbf{Q} = \text{round}((2^N - 1)F(\mathbf{W})), \tag{32}$$

where $F(\cdot) : \mathbb{R} \mapsto [0, 1]$ is a normalization function. In this work, we utilize the 4-bit NormalFloat Quantization (NF4) proposed in [29] to model the normalization function. It assumes $\mathbf{W} \sim N(0, \sigma^2)$ and hence $F(\mathbf{W}) = \Phi(\mathbf{W}/\sigma)$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Besides, other quantization methods can also be involved, such as uniform quantization.

**SVD Step:** After obtaining the $i$-th quantized weight $\mathbf{Q}_i$, SVD is applied to the residual of the quantization, denoted by $\mathbf{R}_i = \mathbf{W} - \mathbf{Q}_i$:

$$\mathbf{R}_i = \sum_{j=1}^{d} \sigma_{i,j}\mathbf{u}_{i,j}\mathbf{v}_{i,j}^T, \tag{33}$$

where $d = \min\{d_1, d_2\}$, $\sigma_{i,1} \geq \sigma_{i,2} \geq \ldots \geq \sigma_{i,d}$ are the singular values of $\mathbf{R}_i$, $\mathbf{u}_{i,j}$, $\mathbf{v}_{i,j}$ are the associated left and right singular vectors of $\mathbf{R}_i$. We then obtain a rank-$r$ approximation of $\mathbf{R}_i$ by $\mathbf{A}_{i-1}\mathbf{B}_{i-1}^T$, where

$$\mathbf{A}_i = [\sqrt{\sigma_{i,1}}\mathbf{u}_{i,1}, \ldots, \sqrt{\sigma_{i,r}}\mathbf{u}_{i,r}], \tag{34}$$

$$\mathbf{B}_i = [\sqrt{\sigma_{i,1}}\mathbf{v}_{i,1}, \ldots, \sqrt{\sigma_{i,r}}\mathbf{v}_{i,r}]. \tag{35}$$

After obtaining the initialization, LoRA fine-tuning can be performed as described in Section III-C.

Moreover, during the inference, we merge the quantized backbone with the finetuned adapters to acquire the final output. If the deployed device is resource-limited, the merged model can be further quantized before deployment; otherwise, the full-precision model can be deployed directly.

**Algorithm 1** The initialization of LoRA fine-tuning-aware quantization.

**Input:** Full-precision pre-trained weight $\mathbf{W}$, LoRA rank $r$, $N$-bit quantization function $q_N(\cdot)$, iteration number $Iter$.

1: Initialize $\mathbf{A}_0 \leftarrow \mathbf{0}, \mathbf{B}_0 \leftarrow \mathbf{0}$;
2: **for** $i = 1$ to $Iter$ **do**
3: Obtain quantized weight $\mathbf{Q}_i \leftarrow q_N(\mathbf{W} - \mathbf{A}_{i-1}\mathbf{B}_{i-1}^T)$;
4: Obtain low-rank approximation $\mathbf{A}_i, \mathbf{B}_i \leftarrow \text{SVD}(\mathbf{W} - \mathbf{Q}_i)$ based on [34]-[35];
5: **end for**

**Output:** $\mathbf{Q}_{Iter}$, $\mathbf{A}_{Iter}$, and $\mathbf{B}_{Iter}$.

### C. Multi-task Loss Function and Training Schedule

During the training stage, we jointly train the selected tasks, and the multi-task loss function is written as

$$\text{Loss} = \sum_n \alpha_n \text{Loss}_n, \tag{36}$$

where $\text{Loss}_n$ is loss function of task $n$, which are linearly combined with task weights $\alpha_n$. In this work, we set $\alpha_{\text{CP}} = 1$, $\alpha_{\text{Det}} = 5$, and $\alpha_{\text{Pre}} = 0.2$. It is worth noting that the backbone of the pre-trained LLM is frozen, while the other parameters of the network, together with the LoRA adapters, are trainable. Then the proposed network updates the parameters of LoRA adapters using the corresponding loss. The detailed loss functions for each task are illustrated below.

For **multi-user precoding**, the loss function can be directly derived from the original optimization objective, i.e., we employ negative sum rate as the optimization objective in an unsupervised learning manner:

$$\text{Loss}_{\text{Pre}} = -\sum_{k=1}^{K} \log_2(1 + \gamma_k). \tag{37}$$

However, as the calculation of the sum rate is complicated, involving many complex matrix operations, both the loss calculation and the corresponding gradient computation would be time-consuming. Thereby, inspired by [30], a two-stage training method can be employed, namely supervised learning and unsupervised learning, respectively. In the supervised learning stage, we can first generate the power allocation vectors $\underline{\mathbf{p}}$ and $\underline{\lambda}$ using the WMMSE algorithm as the label. Then, the supervised learning will employ the MSE loss function to make the power allocation vectors, $\hat{\mathbf{p}}$ and $\hat{\lambda}$, generated by LLM as close to $\underline{\mathbf{p}}$ and $\underline{\boldsymbol{\lambda}}$ as possible, i.e.,

$$\text{Loss}_{\text{Pre}} = \frac{1}{2K}\left(\|\underline{\mathbf{p}} - \hat{\mathbf{p}}\|_2^2 + \|\underline{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}\|_2^2\right). \tag{38}$$

Nevertheless, the WMMSE algorithm achieves only local optimality, making (38) insufficient for fully addressing the fundamental objective of problem **P3**. To enhance the overall rate performance, additional network training is implemented using an unsupervised learning manner.

In the **signal detection** task, the original transmitted data is rearranged as $\mathbf{X}_{\text{Det}} \in \mathbb{R}^{K\times 2}$, which serves as ground truth of the network output. Then we choose MSE loss for the training,

$$\text{Loss}_{\text{Det}} = \frac{1}{2K}\|\mathbf{X}_{\text{Det}} - \mathbf{X}_{\text{Det}}^O\|_F^2. \tag{39}$$

For **channel prediction**, the ground truth of the predicted CSI is also available. We transform the complex CSI matrix to real ground truth $\mathbf{X}_{\text{CP}} \in \mathbb{R}^{T_2\times M\times 2}$, and MSE is adopted as the loss function to minimize the prediction error, i.e.,

$$\text{Loss}_{\text{CP}} = \frac{1}{2MT_2}\|\mathbf{X}_{\text{CP}} - \mathbf{X}_{\text{CP}}^O\|_F^2. \tag{40}$$

## V. Simulation Results

In this section, extensive numerical simulations are presented to verify the effectiveness of the proposed method. Firstly, we elaborate on the simulation setup. Then the performance of three selected tasks, i.e., channel prediction, multi-user precoding, and signal detection, is evaluated respectively. Besides, the performance of LoRA fine-tuning-aware quantization is compared with the full-precision model. Then, we analyze the impact of the designed multi-task instruction and different LLM backbones, respectively. Finally, the few-shot learning and generalization ability of the proposed multi-task LLM are evaluated.

### A. Simulation Setup and Training Details

For the experimental setup, we utilize the QuaDRiGa channel generator [31], implementing the 3GPP Urban Macro (UMa) propagation model [32] under non-line-of-sight (NLOS) conditions. The channel consists of 21 scattering clusters, each containing 20 propagation paths. We consider a multi-user MISO-OFDM system, where a BS simultaneously serves $K = 4 \sim 8$ moving users. BS employs a UPA comprising $N_h = 16$ elements in horizontal and $N_v = 8$ in vertical, while users are configured with single-antenna receivers. The antenna spacing is maintained at half the wavelength at the center frequency. The users are uniformly distributed within angle range $[-\pi/2, \pi/2]$, and distance range $[\rho_{\min}, \rho_{\max}] = [20\text{ m}, 100\text{ m}]$. We suppose a time-division duplex (TDD) system, where the center frequency of the channel is set as $2.4$ GHz. The bandwidth of the channel is 8.64 MHz, comprising $M = 48$ subcarriers, i.e., the frequency interval of subcarriers is 180 kHz. The dataset is partitioned into training and testing subsets, containing 50,000 and 10,000 samples per task, respectively. The model undergoes training for 500 epochs over the dataset. Besides, the rank of LoRA is set as 16, and the batch size is 100.

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2025.3626010



For the channel prediction problem, we predict the future CSI of $T_2 = 4$ time slots based on the historical CSI of $T_1 = 16$ time slots. We suppose that the users are initialized with random positions and follow linear movement patterns. The velocity distribution for mobile users spans uniformly from 10 km/h to 100 km/h. To enhance the robustness against noise, the SNR is uniformly sampled between 0 dB and 20 dB during the training stage, to account for both the low-noise and high-noise scenarios. For uplink signal detection, we suppose the transmitted data is generated from 16-QAM modulation symbol. Similarly, the received signal is corrupted by noise with SNR uniformly distributed between 0 dB and 20 dB during the fine-tuning.

### B. Performance Evaluation for Channel Prediction

*1) Baselines and Performance Metric:* To evaluate the performance, we compare the proposed multi-task LLM with the following benchmarks.

- Transformer: [5] introduces a parallelized channel prediction framework based on transformer to predict future CSI in parallel, and thus avoid error propagation problems.
- RNN: A Recurrent Neural Network (RNN) [33] is a typical neural network used for processing sequences and is commonly utilized in channel prediction tasks. In the experiments, we configure the RNN with four layers.
- LSTM: The long short-term memory network (LSTM) [34] incorporates specialized memory units and gating mechanisms to effectively capture long-range temporal dependencies. Our implementation utilizes a four-layer LSTM structure for predictive modeling.
- GRU: As an enhanced version of LSTM, the gated recurrent unit (GRU) [35] introduces simplified gating operations to mitigate gradient-related challenges during training. Similarly, the GRU-based model utilized in this work consists of 4 layers.
- LLM4CP: LLM4CP [13] represents a pioneering effort in finetuning layer normalization parameters of pre-trained LLM, i.e., GPT-2, for the channel prediction task.
- Single-task LLM: We also train the proposed network on a single task to better compare the performance with the proposed multi-task LLM.

In channel prediction evaluation, the NMSE serves as a crucial indicator for assessing prediction precision, making it an essential measurement criterion in our experiments.

*2) Performance Analysis:* The evaluation dataset for channel prediction comprises 10 distinct velocities spanning from 10 km/h to 100 km/h, with each velocity containing 1000 data samples. As illustrated in Fig. 5, the NMSE performance of our proposed multi-task LLM framework is compared against various baseline methods across different user velocities. The historical CSI data is added by Gaussian white noise with SNR = 20 dB. Experimental results demonstrate a consistent degradation in NMSE performance across all methods as user mobility increases. This phenomenon can be attributed to the accelerated channel variation and reduced coherence time associated with higher velocities, which consequently amplifies the complexity of accurate channel estimation. Fig. 5

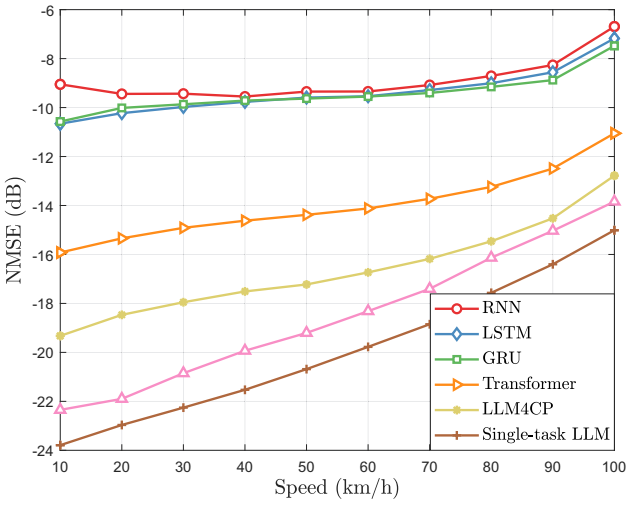


Fig. 5. The NMSE performance of the proposed method and other baselines versus different user velocities.

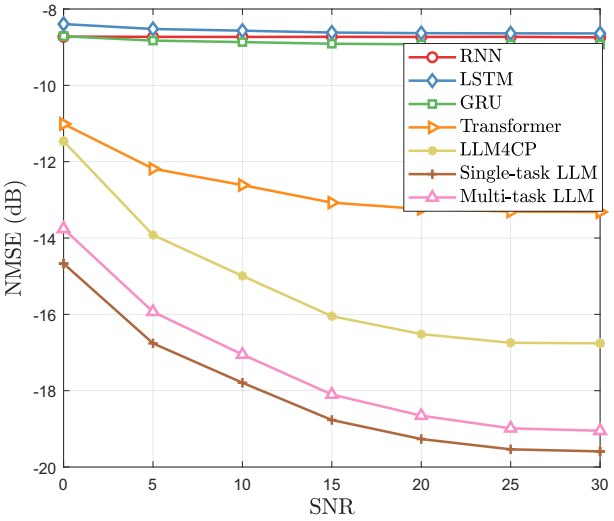


Fig. 6. The NMSE performance of the proposed method and other baselines versus different SNR.

reveals that attention-based methods achieve relatively higher performance than traditional AI methods, validating the potential of attention-based methods in the channel prediction task. Equipped with excellent modeling capability of LLM, the proposed model, finetuned on the channel prediction task only, consistently outperforms other baselines among testing velocities. The proposed single-task LLM achieves better performance than LLM4CP since we apply LoRA finetuning to parameters of both the multi-head attention module and the MLP module in the LLM backbone, except for finetuning parameters of layer normalizations. With elaborately designed multi-task instruction, the proposed LLM-enabled multi-task model obtains comparable channel prediction accuracy with the single-task one, which verifies the effectiveness of the proposed multi-task LLM.

In Fig. 6, the robustness against noise of the proposed method is evaluated, where the SNR of noise in historical

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2025.3626010



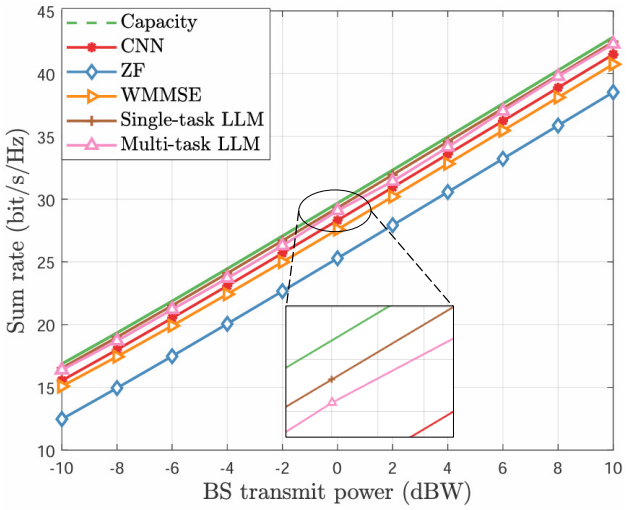


Fig. 7. The sum rate performance of the proposed method and other baselines versus different BS transmit powers.

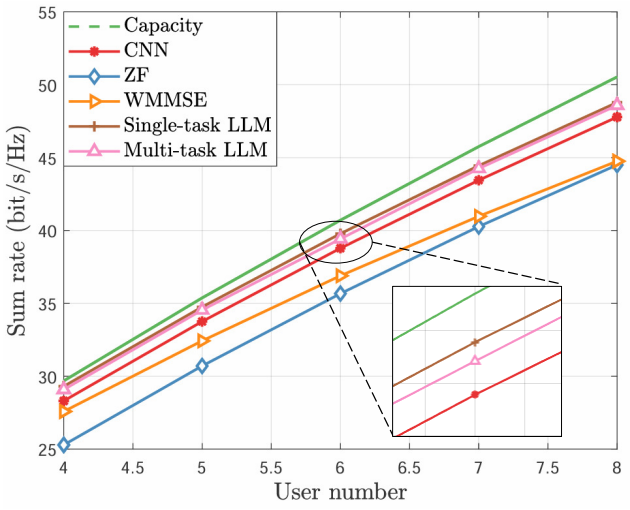


Fig. 8. The sum rate performance of the proposed method and other baselines versus different user numbers.

CSI is growing from 0 dB to 30 dB. The NMSE performance has been averaged over all test speeds. It can be observed that for all schemes, increased SNR conditions lead to improved NMSE performance in prediction accuracy. Thanks to the generalization ability of LLMs, the proposed method exhibits high robustness performance against CSI noise. It achieves the lowest NMSE performance in the entire SNR regime.

### C. Performance Evaluation for Multi-user Precoding

*1) Baselines and Performance Metric:* For multi-user precoding, the following methods are selected as baselines, including traditional methods and deep learning-based methods.

- ZF: Eigen-based zero-forcing (ZF) algorithm [36] is a computationally efficient approach, which derives the precoding matrix through the Moore-Penrose pseudo-inverse operation applied to the channel matrix.
- WMMSE: As mentioned above, the WMMSE algorithm [37] is one of the most popular iterative algorithms. The method can achieve satisfactory sum rate performance while suffering from high computational complexity. The iterative number is set as 20.
- CNN: In [30], the authors propose a CNN-based framework for the optimization of downlink beamforming.
- Single-task LLM: Similarly, we train the proposed network on multi-user precoding only.

The sum rate of users, which is the objective of multi-user precoding, is utilized as the performance metric to evaluate the multi-user precoding task.

*2) Performance Analysis:* The sum rate performance against different BS transmit power is plotted in Fig. 7. The noise power is set as $\sigma^2 = -10$ dBW, the maximal transmit power of BS $P_{\max}$ increases from $-10$ to 10 dBW. Besides, the user number is set as 4. As illustrated in Fig. 7, with the increase of the BS transmit power, the sum rates of all methods increase accordingly. As depicted in Fig. 7, the ZF-based method, though with low complexity, achieves unsatisfactory performance. The iterative algorithm WMMSE improves the sum rate, while it is still possible to fall into local optimal solutions, inducing an obvious gap from the capacity. With the powerful network and carefully designed training strategy, the deep-learning-based methods, including the CNN-based method and LLM-based method, are promising in conquering the problem and further improving the performance. Specifically, the proposed model, both trained on a single task and trained on multiple tasks, achieves near-optimal performance for all transmit power and outperforms other benchmark schemes for the entire transmit power range. Owing to the increasing size of the network, the LLM-based method exhibits superior optimization and generalization capabilities and outperforms the CNN-based method.

In Fig. 8, we illustrate the sum rate performance against the user number, which ranges from 4 to 8. The transmit power and noise power are set as 0 dBW and -10 dBW, respectively. As the user number increases, the spectrum efficiency increases with further exploitation of multiplexing gain. We observe from Fig. 8 that the proposed multi-task LLM-based method enjoys a higher sum rate performance, compared to existing methods for different numbers of users. This verifies the effectiveness and scalability of the proposed method. It is worth noting that LLMs inherently possess the ability to process variable-length sequences; thus, our proposed scheme can be directly applied to different numbers of users without requiring any modifications. In contrast, for traditional AI methods, to accommodate varying numbers of users, input data under different user numbers must be zero-padded to match the shape of the maximum user number.

### D. Performance Evaluation for Signal Detection

*1) Baselines and Performance Metric:* To validate the effectiveness of the proposed method, several methods are implemented as baselines.

- LMMSE: Linear minimum mean-squared error (LMMSE) detector is a classical method for achieving signal detection with low complexity.

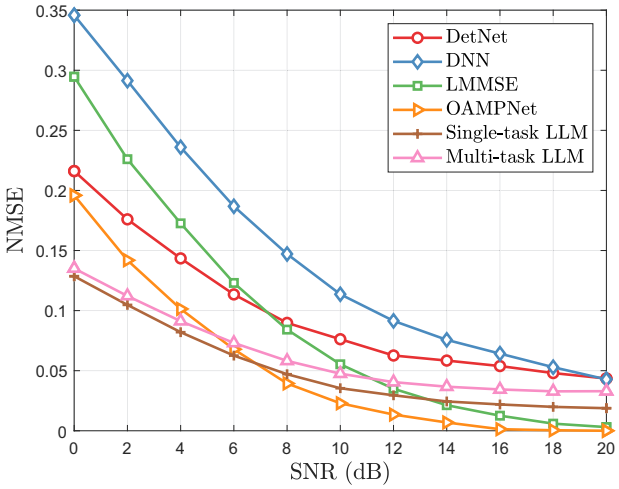


Fig. 9. The NMSE performance of the proposed method and other baselines versus different SNRs.

- DNN: In [38], a data-driven model which inputs the channel as well as the received data and outputs the original data through a deep learning network.
- DetNet: The detection network (DetNet) in [39] unfolds the iterations of a projected gradient descent algorithm to recover the data.
- OAMP-Net: A famous model-driven deep learning network proposed in [6], which incorporates deep learning into the orthogonal AMP (OAMP) algorithm for accurate signal detection.
- Single-task LLM: The proposed network is finetuned only on the signal detection task.

In this work, we utilize NMSE and symbol error ratio (SER) as metrics to evaluate the performance of signal detection. NMSE loss is the direct performance metric to present the accuracy of data recovery. It should be noted that the reason for employing the data recovery accuracy rather than only choosing the bit error ratio (BER) or SER to optimize and evaluate the model is described below. In the practical communication link, the objective of signal detection is to recover the data as accurately as possible, which can be utilized to compute the log-likelihood ratio (LLR) for demodulation and channel decoding to enable reliable communications. Besides, obtaining the recovered signal, we also demodulate the transmitted symbol by minimum Euclidean distance decision, and then SER is employed to evaluate the performance in this case.

*2) Performance Analysis:* In Fig. 9, the NMSE performance of the proposed multi-task LLM on signal detection under different SNRs is presented. We also compare the performance with several baselines. As illustrated in Fig. 9, the DNN-based method presents poor performance since it directly inputs the data to the black-box-based network, while ignoring the domain knowledge and structure of the problem and data. Besides, despite the low computational complexity, the LMMSE-based method also shows relatively high NMSE performance. The model-based methods, including DetNet [39] and OAMP-Net [6], significantly improve the data recovery accuracy; the OAMPNet [6] achieves higher performance than the LLM-based method in the high SNR regime with better utilization of the statistical information of noise. Moreover, the proposed multi-task LLM can accurately recover the transmitted data, especially in the low SNR regime. It can be observed that the performance of multi-task LLM is comparable to that of single-task LLM, suggesting the potential of multi-task LLM networks in wireless communications.

The SER performance against SNR is depicted in Fig. 10. When the SNR is lower than 8 dB, the proposed multi-task LLM network outperforms other baselines, which indicates that the generalization capability of LLMs endows the proposed method with high robustness against noise. As the SNR increases, the OAMPNet [6] achieves the best performance, since it fully utilizes the statistical information of noise. Based on this observation, we provide two comments as follows. First, for the scenarios where the power of noise $\sigma^2$ can be obtained, future works can consider effectively incorporating statistical information of the channel and noise as prompts to further improve the performance of the LLM-based method, especially in high SNR regimes. Secondly, there are still many cases where the statistical information of noise may not be obtained in the practical system. In these cases, OAMPNet may fail to achieve satisfactory performance.

Finally, we present a brief comparison of the performance of single-task LLM and multi-task LLM after the specific illustration of three selected tasks. Due to the increasing complexity of joint optimization and the inherent trade-offs across tasks, single-task LLMs slightly outperform the multi-task LLM. Fortunately, the observed performance gap remains marginal, even with near-identical precoding performance achieved in multi-task implementations versus single-task implementations. On the other hand, the multi-task framework significantly reduces memory usage and deployment overhead. For example, the proposed unified model in this work simultaneously addresses three distinct tasks with merely one-third of the parameter count required by individual task-specific models. Therefore, **this favorable trade-off between model efficiency and task performance** substantiates the feasibility and critical value of multi-task frameworks for practical deployment scenarios.

### *E. Performance evaluation for LoRA-fine-tuning-aware quantization*

For fair and convenient comparison, in the above subsections, we mainly utilize the model based on full-precision LoRA fine-tuning for performance evaluation. Then, in this subsection, we focus on the impact of LoRA-fine-tuning-aware quantization. The performance comparison of the proposed model with full-precision LoRA fine-tuning and the model with LoRA-fine-tuning-aware quantization is presented in Table I. For the full-precision model, the model parameters are stored and computed in a floating-point format. On the contrary, the model with LoRA-fine-tuning-aware quantization transforms the LLM backbone into a 4-bit integer format, which indicates significant storage reduction during fine-tuning. Besides, the iterative number for initialization is set as $Iter = 5$. It is noted here that during LoRA fine-tuning, the quantized weight is temporarily dequantized to

TABLE I
COMPARISON OF THE MODEL WITH LORA-FINE-TUNING-AWARE QUANTIZATION WITH OTHER BASELINES.

| Quantization method | NMSE for CP | NMSE for DET | Sum rate for PRE |
|---|---|---|---|
| Full-precision | -18.98 dB | 0.0336 | 29.3086 bit/s/Hz |
| LoRA-fine-tuning-aware quantization | -18.86 dB | 0.0256 | 29.2343 bit/s/Hz |

• "CP" denotes channel prediction, "DET" denotes signal detection, and "PRE" denotes multi-user precoding.

TABLE II
COMPARISON OF THE MODEL WITH DIFFERENT LLM BACKBONE.

| LLM backbone | NMSE for CP | NMSE for DET | Sum rate for PRE | Trainable/Total parameters |
|---|---|---|---|---|
| GPT-2 | -18.98 dB | 0.0336 | 29.3086 bit/s/Hz | 21.76/145.37 M |
| GPT-2(6) | -18.07 dB | 0.0467 | 28.7653 bit/s/Hz | 20.16/101.73 M |
| - | -14.03 dB | 0.2314 | 27.2334 bit/s/Hz | 18.58/58.08 M |

• "CP" denotes channel prediction, "DET" denotes signal detection, and "PRE" denotes multi-user precoding.

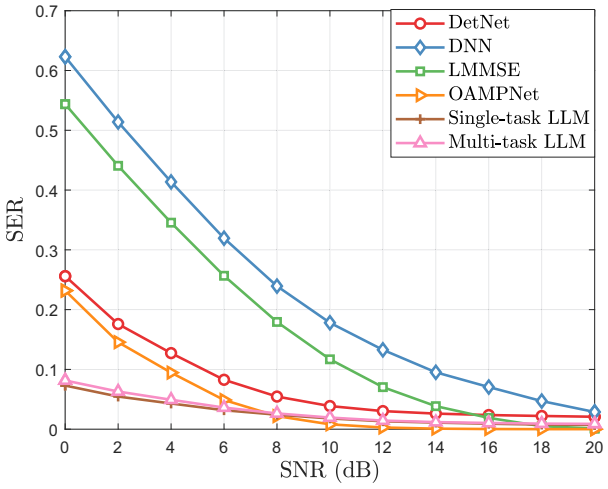


Fig. 10. The SER performance of the proposed method and other baselines versus different SNRs.

the simulated high-precision weight (16-bit floating-point) to facilitate accurate computation.

The results are shown in Table I. For the channel prediction task, the NMSE is averaged for different speeds and SNRs; for the signal detection task, the performance is averaged for different SNRs; for the multi-user precoding task, we set user number as 4, the power of noise and transmit power are set as -10 dBW and 0 dBW, respectively. As shown in Table I, the model with LoRA-fine-tuning-aware quantization achieves comparable performance with the proposed full-precision model. We can conclude from the results that the introduced LoRA-fine-tuning-aware quantization successfully approximates high-precision weights by the quantized weights and low-rank adapters, and thus the performance degradation resulting from quantization is negligible.

Next, we provide quantitative analysis of the reduced resource consumption brought by LoRA and quantization, from aspects including training time, and trainable/total parameters. The proposed model without LoRA fine-tuning contains 143.02 M parameters, consisting of 124.44 M parameters in the GPT-2 backbone and 18.58 M parameters in other modules. If full-parameter fine-tuning is directly employed, the trainable parameters are 143.02 M. Based on the proposed LoRA fine-tuning strategy, the trainable parameter number in the GPT-2 backbone drops sharply from 124.44 M to 3.18 M, while the total trainable parameter is 21.76 M. The significantly reduced trainable parameters can reduce the training time, computational complexity, as well as GPU memory usage. Specifically, since training/inference time and GPU memory vary across different GPU/CPU configurations, sequence lengths, batch sizes, etc, we report the comparison on the channel prediction task with batch size 100 as an example using NVIDIA GeForce 24G RTX4090 GPUs. Compared with full-parameter finetuning, the training time per batch of LoRA fine-tuning reduces from 72 ms to 56 ms.

Furthermore, though the involvement of LoRA-fine-tuning-aware quantization may not affect the trainable and total parameter number, it significantly reduces the memory usage for the LLM backbone. As mentioned before, quantization of the LLM backbone from float format into a 4-bit integer format introduces significant storage reduction during fine-tuning. It is worth noting that the advantages introduced by both LoRA and quantization will be more significant and essential as the size of the LLM backbone further increases.

### *F. Impact of multi-task instruction*

In this subsection, we analyze the influence of the proposed multi-task instruction module in Section III-A on the performance and convergence rate of neural network training. It is worth noting that, in order to distinguish different tasks, utilizing multi-task instruction (especially the task identifier part) as prompts is indispensable for the proposed multi-task PHY network to understand different task requirements; besides, the domain knowledge in the task description and instruction part may accelerate convergence. For single-task LLMs, although it is possible to neglect the prompt, introducing the designed instruction with wireless data can involve domain knowledge to facilitate task-specific adaptation of LLMs.

In Fig. 11, we take the signal detection task, for instance, to illustrate the training loss in MSE against the training epoch.

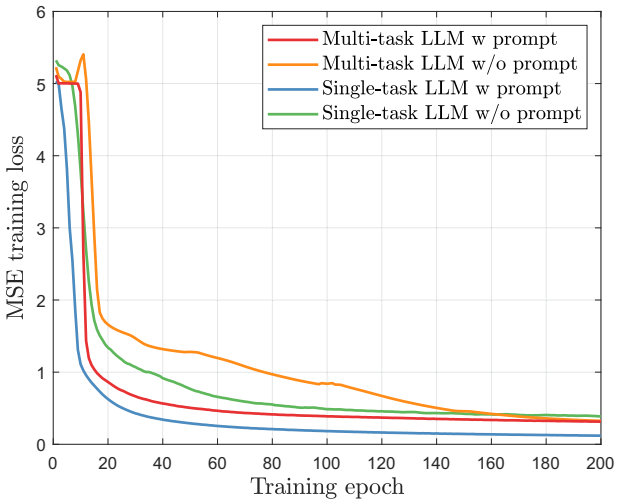


Fig. 11. Training losses of different methods against epoch.

In this figure, we depict the training loss of the proposed model trained for multiple tasks with prompts, the proposed model trained for multiple tasks with only task identifier in prompts, the proposed model trained for a single task with prompts, and the proposed model trained for a single task without prompts in the first 200 epochs. We can observe that the overall trends of all losses are declining with the increase of training epochs. The training loss of single-task LLM converges faster than its multi-task counterpart, due to the complete optimization focus on a single task. Furthermore, the proposed model with designed prompts achieves satisfactory performance faster than the proposed model without prompts. Therefore, it is indicated that the incorporation of designed multi-task instruction as prompts significantly accelerates the network fine-tuning and improves the LLM's adaptability to downstream tasks.

### G. Impact of LLM backbone

Finally, in this subsection, the influence of the LLM backbone on the performance is evaluated. In this work, we employ GPT-2 as the backbone. To analyze the impact, we compare the selected backbone with the following benchmarks. Firstly, we employ the first six transformer blocks of GPT-2 as the backbone, while other modules remain the same. The method is denoted as "GPT-2(6)". Secondly, the LLM backbone is directly removed, which means that the output of the encoders is directly fed into the designed decoders.

We summarize the simulation results in Table II. Similarly, as stated in Section V-E, the NMSE is averaged within different SNRs and speeds for channel prediction and is averaged within different SNRs for signal detection. Besides, user number is set as 4, the power of noise and transmit power are set as -10 dBW and 0 dBW for multi-user precoding. It can be observed from Table II that the performance achieved in different tasks improves with the introduction of the LLM backbone. For instance, the NMSE performance for channel prediction of the proposed method with GPT-2 backbone is 18.98 dB, while the performance drops 5 dB when removing the backbone. Furthermore, due to the increasing size of the LLM backbone, the model with complete GPT-2 backbone outperforms the model with only the first six transformer blocks of the GPT-2 backbone, although it can still achieve the multi-task PHY network. Therefore, we can adopt a proper LLM backbone to balance the computational costs and performance.

### H. Few-shot learning capability

The few-shot learning capability allows deep learning models to achieve strong performance with very limited training data, and thus empowers efficient and swift real-world applications. It is indispensable for wireless communication deployments where massive wireless data collection and network training are costly or even impractical. In this part, we evaluate the few-shot learning ability of the proposed methods. We utilize 10% dataset(i.e., 5000 samples), respectively, to train the proposed model and other baselines.

In Table III, we provide the NMSE performance of the channel prediction task of models trained on the full dataset and 10% dataset for comparison. The results are averaged over all test SNRs and speeds. Pre-trained on extensive datasets, LLMs acquire rich general knowledge, eliminating the need for training from scratch for downlink tasks. By fine-tuning only minimal parameters via LoRA with limited channel prediction data samples, LLMs demonstrate strong few-shot learning capabilities without significant overfitting despite their scale. Experimental results confirm that LLM-based approaches, including LLM4CP, proposed single-task LLM and multi-task LLM, outperform conventional DL models even when trained on merely 10% of the dataset. Besides, the proposed single-task and multi-task LLMs achieve the most accurate channel prediction for all different data numbers.

### I. Generalization Experiments

Generalization ability is also crucial for models to deploy in real-world scenarios. It denoted the capability of models to maintain performance in new communication scenarios or against noise without retraining. To illustrate the generalization ability, we provide two case studies. First, for multi-user precoding, we assume accurate multi-user channel can be obtained. However, the estimated channel may be corrupted by noise, and thus the beamformers can only be acquired based on inaccurate multi-user channel. Therefore, the robustness of the proposed method against noise is evaluated with imperfect channel for multi-user precoding. Secondly, in order to elaborate on the cross-scenario generalization ability, we directly apply the model trained in the UMa scenario to the 3GPP Urban Micro (UMi) scenario without any additional training process for the channel prediction task.

For multi-user precoding, the sum rate performance of the proposed method and other baselines versus SNR of the estimated channel is presented in Fig. 12. The user number is set as 4, and the transmit power and noise power are set as 0 dBW and -10 dBW, respectively. As the SNR of the estimated channel increases from 0 dB to 30 dB, the sum rates of all methods increase. The advantages of the proposed multi-task LLM over other baselines are even more evident

TABLE III
FEW-SHOT LEARNING PERFORMANCE OF CHANNEL PREDICTION TASK (NMSE IN DB).

| Dataset | Multi-task LLM | Single-task LLM | LLM4CP | Transformer | RNN | LSTM | GRU |
|---|---|---|---|---|---|---|---|
| Full | -18.98 | -19.33 | -15.65 | -12.80 | -8.75 | -8.11 | -8.32 |
| 10% | -13.96 | -14.74 | -11.36 | -9.39 | -7.80 | -5.96 | -7.28 |

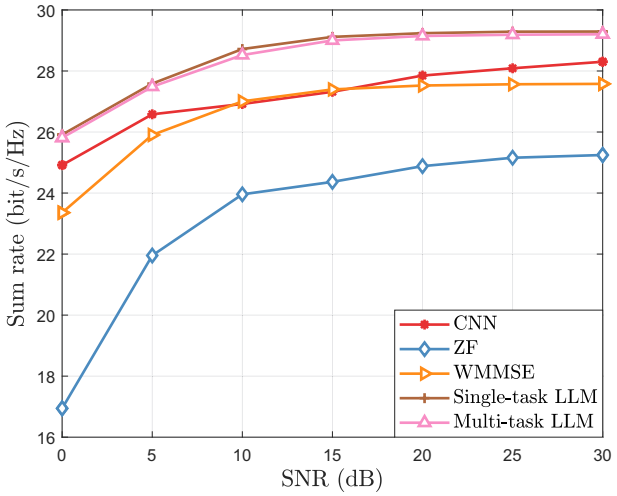


Fig. 12. The sum rate performance of the proposed method and other baselines with imperfect multi-user channel for multi-user precoding.

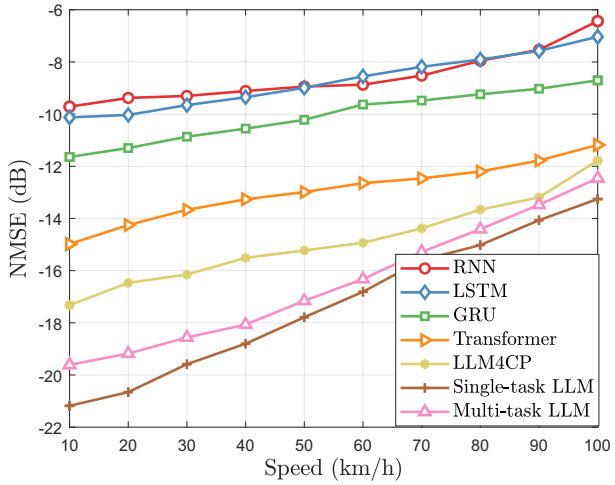


Fig. 13. The zero-shot generalization performance for the channel prediction in the UMi scenario.

in low SNR regions, indicating the superior robustness of the proposed multi-task LLM against noise. Fig. 13 elaborates the NMSE performance of channel prediction in the UMi scenario with the model trained in the UMa scenario. Except for the scenarios, we keep the other settings unchanged. The proposed model surpasses other baselines in terms of the NMSE metric, demonstrating its strong generalization capability across different channel distributions.

## VI. CONCLUSIONS

In this paper, we propose an LLM-enabled multi-task PHY network to unify multiple tasks with a single LLM. Multi-task instruction module, input encoders, as well as output decoders, are elaborately designed to distinguish multiple tasks and adapt the features of different formats of wireless data for the feature of LLM. Moreover, to reduce the memory requirements of the proposed model, a LoRA fine-tuning-aware quantization method is introduced. Simulation results have verified the effectiveness of the proposed method. The proposed LLM framework is promising to perform different tasks using a single model, significantly saving the redundancy and costs of the practical deployment of LLM. It makes an initial attempt to provide a more adaptable, comprehensive, and intelligent PHY network with the aid of LLMs. Future works can be focused on incorporating more tasks into the network. Besides, further improvement of LoRA and quantization specifically designed for the wireless domain can be considered for subsequent research. Moreover, implementing a unified data encoder/decoder framework to extract task-discriminative features across diverse tasks could further enhance architectural consistency in the multi-task LLM framework.

## REFERENCES

[1] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-native air interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.
[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cog. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[3] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.
[4] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
[5] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate channel prediction based on transformer: Making mobility negligible," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2717–2732, Sep. 2022.
[6] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Sigal Process.*, vol. 68, pp. 1702–1715, Feb. 2020.
[7] M. K. Shehzad, L. Rose, M. M. Butt, I. Z. Kovács, M. Assaad, and M. Guizani, "Artificial intelligence for 6G networks: Technology advancement and standardization," *IEEE Veh. Tech. Mag.*, vol. 17, no. 3, pp. 16–25, May 2022.
[8] R. K. Alec, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
[9] H. Touvron, L. Martin, K. R. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, and *et al.*. Shruti Bhosale, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and *et al.*. Amanda Askell, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020, pp. 1–5.
[11] W. Lee and J. Park, "LLM-empowered resource allocation in wireless communications systems," *arXiv preprint arXiv:2408.02944*, 2024.
[12] J. Guo, Y. Cui, C.-K. Wen, and S. Jin, "Prompt-enabled large AI models for CSI feedback," *arXiv preprint arXiv:2501.10629*, 2025.
[13] B. Liu, X. Liu, S. Gao, X. Cheng, and L. Yang, "LLM4CP: Adapting large language models for channel prediction," *J. Commun. Inf. Netw.*, vol. 9, no. 2, pp. 113–125, Jun. 2024.

[14] Y. Sheng, K. Huang, L. Liang, P. Liu, S. Jin, and G. Y. Li, "Beam prediction based on large language models," *arXiv preprint arXiv:2408.08707*, 2024.

[15] J. Shao, J. Tong, Q. Wu, W. Guo, Z. Li, Z. Lin, and J. Zhang, "WirelessLLM: Empowering large language models towards wireless intelligence," *J. Commun. Info. Netw.*, vol. 9, no. 2, pp. 99–112, Jun. 2024.

[16] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, and *et al.*. Wu, Di, "Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Commun. Surv. Tutor.*, Sep. 2024.

[17] W. Yu, H. He, S. Song, J. Zhang, L. Dai, L. Zheng, and K. B. Letaief, "AI and deep learning for THz ultra-massive MIMO: From model-driven approaches to foundation models," *arXiv preprint arXiv:2412.09839*, 2024.

[18] F. Jiang, Y. Peng, L. Dong, K. Wang, K. Yang, C. Pan, D. Niyato, and O. A. Dobre, "Large language model enhanced multi-agent systems for 6G communications," *IEEE Wireless Communications*, vol. 31, no. 6, pp. 48–55, Dec. 2024.

[19] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, "Large generative AI models for telecom: The next big thing?" *arXiv preprint arXiv:2306.10249*, 2023.

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[21] Y. Li, Y. Yu, C. Liang, P. He, N. Karampatziakis, W. Chen, and T. Zhao, "LoftQ: LoRA-Fine-Tuning-Aware Quantization for large language models," *arXiv preprint arXiv:2310.08659*, 2023.

[22] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, Jul. 2014.

[23] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *J. Commun. Netw.*, vol. 15, no. 4, pp. 338–351, Aug. 2013.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and *etc.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *2021 International Conference on Learning Representations (ICLR)*, 2021.

[25] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[27] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, and C. L. *et al.*, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[28] W. Qi, Y.-P. Ruan, Y. Zuo, and T. Li, "Parameter-efficient tuning on layer normalization for pre-trained language models," *arXiv preprint arXiv:2211.08682*, 2022.

[29] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," *arXiv preprint arXiv:2305.14314*, 2023.

[30] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Dec. 2020.

[31] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas and Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.

[32] 3GPP RADIO ACCESS NETWORK WORKING GROUP, "Study on channel model for frequencies from 0.5 to 100 GHz(Release 15)[R]." 3GPP TR 38.901, 2018.

[33] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, Aug. 2019.

[34] ——, "Deep learning for fading channel prediction," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, Mar. 2020.

[35] I. Helmy, P. Tarafder, and W. Choi, "LSTM-GRU model-based channel prediction for one-bit massive MIMO system," *IEEE Trans. Veh. Tech.*, vol. 72, no. 8, pp. 11 053–11 057, Mar. 2023.

[36] C. Zhang, Y. Jing, Y. Huang, and L. Yang, "Performance analysis for massive MIMO downlink with low complexity approximate zero-forcing precoding," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3848–3864, Sep. 2018.

[37] S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.

[38] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Sep. 2018.

[39] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. 18th IEEE Int. Workshop Signal Process. Advances Wireless Commun (SPAWC)*, Hokkaido, Japan, Jul. 2017, pp. 1–5.


**Tianyue Zheng** (Graduate Student Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2022. She is currently pursuing the Ph.D. degree in the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her research interests include extremely large scale MIMO (XL-MIMO), CSI acquisition and AI for communications. She has received the National Scholarship in 2019 and the Excellent Student of Jiangsu Province in 2021.

**Linglong Dai** (Fellow, IEEE) received the B.S. degree from Zhejiang University, Hangzhou, China, in 2003, the M.S. degree from the China Academy of Telecommunications Technology, Beijing, China, in 2006, and the Ph.D. degree from Tsinghua University, Beijing, in 2011. From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electronic Engineering, Tsinghua University, where he was an Assistant Professor from 2013 to 2016, an Associate Professor from 2016 to 2022, and has been a Professor since 2022. His current research interests include massive MIMO, reconfigurable intelligent surface (RIS), millimeter-wave and Terahertz communications, near-field communications, machine learning for wireless communications, and electromagnetic information theory.

He has coauthored the book *MmWave Massive MIMO: A Paradigm for 5G* (Academic Press, 2016). He has authored or coauthored over 100 IEEE journal papers and over 60 IEEE conference papers. He also holds over 20 granted patents. He has received five IEEE Best Paper Awards at the IEEE ICC 2013, the IEEE ICC 2014, the IEEE ICC 2017, the IEEE VTC 2017-Fall, the IEEE ICC 2018, and the IEEE GLOBECOM 2023. He has also received the Tsinghua University Outstanding Ph.D. Graduate Award in 2011, the Beijing Excellent Doctoral Dissertation Award in 2012, the China National Excellent Doctoral Dissertation Nomination Award in 2013, the URSI Young Scientist Award in 2014, the IEEE Transactions on Broadcasting Best Paper Award in 2015, the Electronics Letters Best Paper Award in 2016, the National Natural Science Foundation of China for Outstanding Young Scholars in 2017, the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2017, the IEEE ComSoc Asia-Pacific Outstanding Paper Award in 2018, the China Communications Best Paper Award in 2019, the IEEE Access Best Multimedia Award in 2020, the IEEE ComSoc Leonard G. Abraham Prize in 2020, the Distinguished Young Scholar of NSFC in 2023, and the IEEE ComSoc Stephen O. Rice Prize in 2025. He has been recognized as a Highly Cited Researcher by Clarivate from 2020 to 2024. He was elevated as an IEEE Fellow in 2022.