A Generative Pre-Trained Language Model for Channel Prediction in Wireless Communications Systems

Bo Lin[1], **Huanming Zhang**[1], **Yuhua Jiang**[1], **Yucong Wang**[1], **Tengyu Zhang**[1], **Shaoqiang Yan**[1,2], **Hongyao Li**[1], **Yihong Liu**[1], **Feifei Gao**[1]
[1]Department of Automation, Tsinghua University, Beijing, China
[2]High-Tech Institute of Xi'an, Xi'an, China

Abstract

Channel prediction can greatly reduce the pilot overhead and is a critical technology in the fifth-generation (5G) and the coming 6G wireless communications systems. Conventional model-based channel prediction methods suffer from limited accuracy due to imperfect temporal modeling, while existing AI-based methods suffer from limited generalization due to inadequate training strategies. Recently, large language models (LLMs) have demonstrated remarkable generalization and generation capabilities across diverse domains such as computer vision, quantitative economics, and bioinformatics, which motivates us to apply LLMs in channel prediction. In this paper, we formulate the 'channel sentence' based on channel correlation, where the channel is regarded as a 'word'. Subsequently, we propose a **g**enerative **p**re-**t**rained language model for **c**hannel **p**rediction (CP-GPT). We collect 12M channel data according to the 3GPP 38.901 protocol and train CP-GPT based on the transformer decoder architecture. Moreover, we design two pre-training tasks based on the characteristics of wireless channels to enhance CP-GPT's understanding of communications channels. We further propose a comprehensive benchmark to rigorously evaluate the capabilities of CP-GPT across multiple dimensions. The simulation results demonstrate that CP-GPT has successfully learned various channel characteristics and exhibits impressive capabilities across numerous downstream tasks.

1 Introduction

With the rapid advancement of artificial intelligence technologies, numerous data-driven applications have emerged in areas like smart cities, intelligent manufacturing, autonomous driving, and remote healthcare, in which massive amounts of data need to be exchanged among ubiquitous edge devices. To accommodate vast data traffic, communications systems should support ultra-high data rates, for which the prerequisite is the accurate acquisition of the wireless channels (Khan et al., 2022). In general, the acquisition of channels is based on transmitting substantial pilots (Manasa and Venugopal, 2024). When the data volume increases, the pilot overhead often grows proportionally. Therefore, efficient channel prediction (Zeng et al., 2024; Kim et al., 2020) methods are often adopted to reduce the pilot overhead.

Existing channel prediction methods can be primarily categorized into model-based channel prediction and AI-based channel prediction. Model-based channel prediction methods typically rely on time-series modeling such as autoregressive, to capture channel variations. However, model-based approaches cannot provide satisfactory performance due to idealized assumptions in channel model (Ozawa et al., 2015). On the other side, AI-based channel prediction methods excavate temporal patterns in channel variations (Stenhammar et al., 2024) to predict future channels. However, existing AI-based approaches exhibit critical limitations in generalization as they are typically trained under specific communications scenarios, which leads to significant performance degradation when deployed in new environments.

Recently, large language models (LLMs) (Zhou et al., 2024a) have demonstrated remarkable generalization and generation capabilities across diverse domains such as computer vision (CV), quantitative economics, and bioinformatics, which motivates us to explore LLMs for channel prediction. In fact, at the premier global event 2025 Mobile World Congress (MWC25) (GSMA, 2025), several well-known organizations, including Huawei and ZTE, presented their outlooks on the deep integration of LLMs with communications systems such as LLM-aided semantic communications and LLM-aided communications networks, which indicates strong potential for LLMs' application in communications systems.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13406–13419
November 4-9, 2025

In this paper, we conceptualize wireless channels as the inherent 'word' of communications systems and propose a generative pre-trained language model for channel prediction (CP-GPT). The main contributions include:

- **We propose a novel channel-to-language analogy and design an LLM-based framework for channel prediction:** Since the relationship between wireless channels and language has not yet been explicitly investigated, we propose to treat the channel as a 'word' in the context of wireless communications. We introduce the concept of an end-of-channel-sentence (EOCS) to construct 'channel sentences' based on channel correlation. Furthermore, we design a framework to train an LLM for channel prediction.

- **We design two pre-training tasks to enhance channel understanding:** Based on a comprehensive knowledge of wireless communications, we design two specific pre-training tasks: the next channel prediction (NCP) task and the masked channel reconstruction (MCR) task. The NCP task enhances the network's understanding of the time-varying characteristics, while the MCR task enhances the representation ability of the channel embedding layer.

- **We propose a benchmark to evaluate the performance of CP-GPT:** We propose a comprehensive benchmark to assess the performance of the LLM in channel prediction, which includes five channel-related tasks: few-shot channel prediction, channel prediction under different time intervals, channel prediction under different antenna numbers, cross-antenna channel prediction, and cross frequency channel prediction.

2 Related Work

2.1 Channel prediction

Channel prediction is a promising technology aimed at reducing the channel acquisition overhead by predicting future channels based on historical channels. Existing studies on channel prediction can be categorized into two main approaches: model-based methods (Ozawa et al., 2015; Löschenbrand et al., 2023; Kim et al., 2020; Wang et al., 2023; Huang et al., 2024; Yin et al., 2020) and AI-based methods (Jiang and Schotten, 2019; Helmy et al., 2023; Jiang and Schotten, 2020; Jiang et al., 2022; Zhou et al., 2022).

2.1.1 Model-based channel prediction

Model-based channel prediction mainly relies on mathematical modeling and statistical signal processing. Specifically, model based channel prediction first builds a parametric model for the channel sequences, and then estimates the model's parameters from past channels. Next it uses the established model to predict future channels. For instance, autoregressive (AR) modeling predicts future channels by representing each channel as a weighted sum of previous channels, where the weights are estimated from historical channels (Ozawa et al., 2015). The Wiener filter utilizes frequency-domain analysis and statistical channel models to minimize mean-square prediction error of channels (Löschenbrand et al., 2023). The Kalman filter leverages state-space models and updates channel prediction results recursively by fusing model predictions with real-time observations (Kim et al., 2020). However, model-based channel prediction methods have two main limitations: (i) model-based channel prediction typically relies on explicit assumptions about the temporal evolution of wireless channels, yet real-world channels are often governed by complex and unknown dynamics that cannot be accurately captured by such models; (ii) the estimation of model parameters usually involves high computational complexity, especially in 5G and 6G systems with large antenna arrays.

2.1.2 AI-based channel prediction

AI-based methods usually predict channels using temporal neural networks, particularly recurrent neural network (RNN) (Elman, 1990) and long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) network, which have demonstrated remarkable success in capturing temporal dependencies and patterns. In (Jiang and Schotten, 2019), the authors apply an RNN to build a frequency domain channel predictor for wideband communications. In (Helmy et al., 2023), the authors propose a channel prediction scheme based on LSTM model to compensate for the negative effects of imperfect channels, which can improve the system secrecy performance in high mobility scenarios. However, existing AI-based channel prediction schemes often lack generalization ability due to limited training strategies. Recently, the

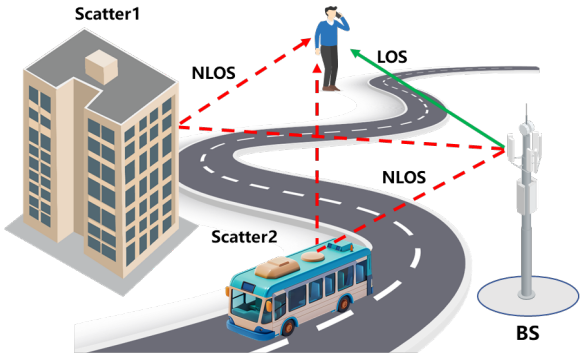


Figure 1: The wireless channel based on multiple propagation paths.

LLM has demonstrated remarkable generation and generalization capabilities in fields of NLP and CV, which inspires us to analogize channel prediction with language modeling.

2.2 LLMs in Communications Systems

Recently, large language models have been preliminarily used in communications systems for semantic transmission (Xie et al., 2024; Guo et al., 2025; Hong et al., 2024; Ni et al., 2025; Jiang et al., 2024; Tang et al., 2024; Yang et al., 2024) and network optimization (Lee and Park, 2024; Liu et al., 2024; Patel et al., 2024; Noh et al., 2025; Sun et al., 2024; Habib et al., 2025; Zhou et al., 2024b).

2.2.1 LLM-aided semantic communications

In (Xie et al., 2024), the authors leverage LLMs to extract semantic information from different modalities of transmitted content. By effectively exchanging the semantics behind the data, the transmission efficiency can be significantly improved. In (Jiang et al., 2024), the authors employ CoDi (Tang et al., 2024) to transform multimodal data into text, which enables semantic transmission rooted in textual information.

2.2.2 LLM-aided network optimization in communications systems

Network optimization commonly involves highly nonconvex and combinatorial problems. Traditional solutions for network optimization are often computationally intensive and may not yield globally optimal solutions. By leveraging the strong mathematical reasoning and generalization abilities of LLMs, it becomes possible to discover more efficient or near-optimal solutions for network optimization. In (Lee and Park, 2024), the authors show that carefully crafted prompts allow the LLM to address resource allocation challenges in communications systems and improve system efficiency. In (Liu et al., 2024), the authors model the communications resource allocation problem as an optimization problem and utilize LLMs to solve the optimization problem. While LLMs have achieved notable results in semantic communications and network optimization, their potential for channel prediction has yet to be explored.

3 Channel Prediction

In this section, we introduce the background knowledge of wireless communications and the definition of channel prediction task.

3.1 Wireless Channel

In wireless communications systems, signals transmitted from the base station (BS) experience various propagation phenomena as shown in Figure 1, including line-of-sight (LOS) transmission, reflection, diffraction, and scattering before arriving at the receiver (Rappaport, 2002). Hence, the received signals often differ considerably from the original transmitted signals. Denote the transmit signal as $\mathbf{X}$, then the received signal is

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}, \tag{1}$$

where $\mathbf{H}$ represents the channel that characterizes the process of signal propagation and $\mathbf{N}$ represents the noise. In order to accurately recover the original signal $\mathbf{X}$ at the receiver, the value of $\mathbf{H}$ should be acquired. Theoretically, the channel $\mathbf{H}$ can be modeled using electromagnetic theory (Wang et al., 2025) as described in Appendix A. However, in practical scenarios, obtaining the channel parameters through electromagnetic computations is virtually impossible due to extremely high computational complexity and significant amount of time required for such calculations. Hence, rather than electromagnetic calculations, practical systems typically adopt pilot-based channel estimation approaches.

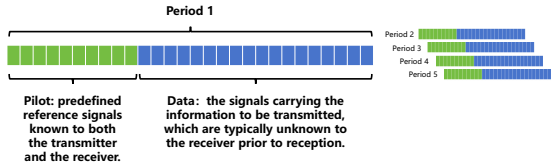
3.2 Channel Estimation

Channel estimation is typically implemented by transmitting known *pilot signals* alongside data (Harkat et al., 2022) as shown in Figure 2(a). The process is described as follows:

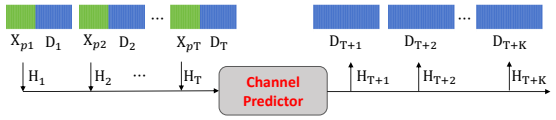
- The transmitter periodically sends predefined pilot signals $\mathbf{X}_p$.

Concept	Wireless Communication	Natural Language
Basic unit	Channel instance ($\mathbf{H}_t$)	Word
Sequence	Channel sequence ($\mathbf{H}_1, \ldots, \mathbf{H}_T$)	Sentence
Context	Communication environment	Linguistic context (background)
Information convey	User message/data	Intended meaning/content
Variation/dynamics	Fading, mobility	Style, tone, expression
Uncertainty	Interference, noise	Ambiguity, linguistic noise

Table 1: Analogy between wireless communications and natural language.



(a) Transmit signals including pilot and data. The green part represents the pilots, and the blue part represents the data to be transmitted.



(b) Illustration of channel prediction, where $\mathbf{X}_{pt}$ denotes the pilot, $\mathbf{H}_t$ denotes the channel, and $\mathbf{D}_t$ denotes the data.

Figure 2: The data structure in wireless communications and the channel prediction pipeline.

- The receiver obtains the received pilots $\mathbf{Y}_p$.
- The channel $\mathbf{H}$ is estimated as:

$$\hat{\mathbf{H}} = \mathbf{Y}_p \mathbf{X}_p^{-1}. \tag{2}$$

- The transmitter sends the data $\mathbf{D}$, which is unknown to the receiver. After transmission over the channel, the receiver observes the received signal $\mathbf{Y}_d$.
- Within an extremely short time interval, the channel can be regarded as quasi-static. Thus, the channel experienced by the transmitted data $\mathbf{D}$ is assumed to be identical to that experienced by the transmitted pilot $\mathbf{X}_p$.
- The receiver utilizes the estimated channel $\hat{\mathbf{H}}$ to recover the transmitted signals:

$$\hat{\mathbf{D}} = \hat{\mathbf{H}}^{-1} \mathbf{Y}_d, \tag{3}$$

where $\hat{\mathbf{D}}$ represents the estimate of the transmitted data. The accuracy of $\hat{\mathbf{D}}$ directly depends on the accuracy of the channel estimate $\hat{\mathbf{H}}$. A more accurate channel estimate leads to smaller errors in $\hat{\mathbf{D}}$, thereby reducing symbol error rates and supporting higher data transmission rates.

In each period, the receivers perform the above channel estimation and signal recovery process. Since wireless channels typically vary over time (Yadav et al., 2021), pilot signals $\mathbf{X}_p$ need to be inserted in each period to enable accurate channel estimation at the receiver. However, as the demand for data transmission continues to grow, frequently inserting a large number of pilot signals $\mathbf{X}_p$ can significantly reduce actual data transmission rate. Therefore, if future channels can be predicted based on past channels, the pilot overhead can be reduced, thereby increasing data transmission rates.

3.3 Channel Prediction Task Definition

As shown in Figure 2(b), channel prediction utilizes the past channels $(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_T)$ to predict the future channel $\mathbf{H}_{T+1:T+K}$, which can be mathematically represented as

$$P(\mathbf{H}_{T+1}, \ldots, \mathbf{H}_{T+K} \mid \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_T). \tag{4}$$

Through channel prediction, the pilot overhead can be reduced by $\frac{K}{K+T}$.

4 The Generative Pre-Trained Language Model for Channel Prediction (CP-GPT)

In this section, we introduce an analogy framework between wireless channels and natural language and present a detailed description of the proposed CP-GPT.

4.1 Analogy between Wireless Channels and Natural Language

In natural language, humans convey their thoughts and intentions through words. In wireless communications systems, users exchange information

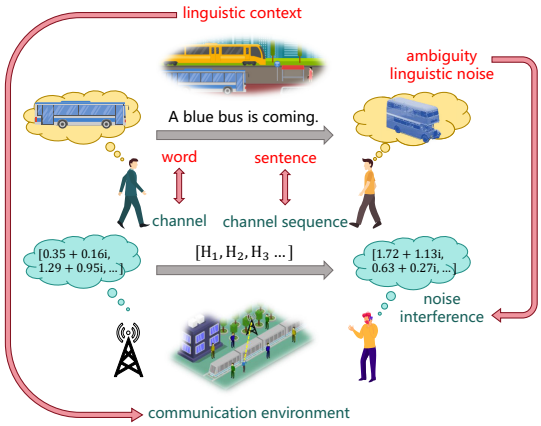


Figure 3: The analogy between wireless communications and natural language.

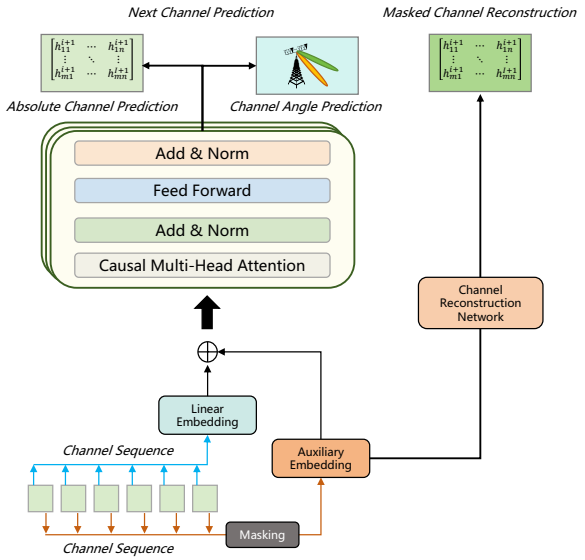


Figure 4: The structure of the proposed CP-GPT.

over wireless channels. Therefore, we can analogize channels to 'words' in wireless communication systems.

In natural language, a sentence has several words where each word is arranged in a specific order to form a meaningful information flow. Similarly, a sequence of channels over time characterizes the evolution of wireless propagation. Therefore, we can analogize the channel sequences to 'sentences'.

Moreover, in natural language, the context or background settings in which a sentence is constructed play a crucial role in shaping the meaning of the words and sentences. Similarly, the physical environment including obstacles, scatterers, and user mobility directly affects the characteristics and thr evolution of channel. Therefore, the physical environment can be analogized to linguistic context in natural language. A comprehensive summary of the analogy between wireless channels and natural language is provided in Table 1 and Figure 3.

Based on the analogy between wireless channels and natural language, we then design the CP-GPT using the LLM.

4.2 CP-GPT

As shown in Figure 4, the CP-GPT includes three parts: channel embedding, transformer decoder, and channel reconstruction network. Since channel is naturally a vector, there is no need to adopt a tokenizer. However, in different environments, the channel amplitudes often exhibit significant variations. To enable stable training across diverse environments, we normalize the channel sequence.

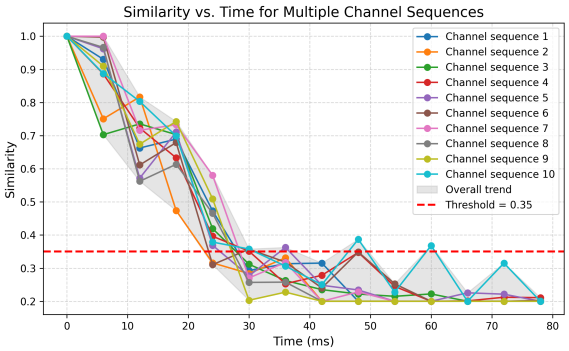


Figure 5: The similarity curve of the channel sequences.

4.3 Channel Normalization

Given a sequence of T channel matrices $\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_T\}$, where each $\mathbf{H}_t \in \mathbb{C}^{N_r \times N_t}$, the normalization process is:

1. Compute the Frobenius norm of $\mathbf{H}_1$:

$$\|\mathbf{H}_1\|_F = \sqrt{\sum_{i=1}^{N_r} \sum_{j=1}^{N_t} |h_{ij}^{(1)}|^2}. \quad (5)$$

2. Normalize all channels by $\mathbf{H}_1$'s norm:

$$\tilde{\mathbf{H}}_m = \frac{\mathbf{H}_m}{\|\mathbf{H}_1\|_F}, \quad \forall m = 1, \ldots, M. \quad (6)$$

4.4 End of Channel Sequence

Similar to word correlations in sentences, adjacent channel matrices exhibit strong temporal correlations. Mathematically, as for the channel sequence

$\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_T\}$, the correlation coefficient between $\mathbf{H}_t$ and $\mathbf{H}_{t+\Delta t}$ is:

$$\rho(\Delta t) = \frac{\mathbb{E}[\text{vec}(\mathbf{H}_t)^H \text{vec}(\mathbf{H}_{t+\Delta t})]}{\|\mathbf{H}_t\|_F \|\mathbf{H}_{t+\Delta t}\|_F}. \quad (7)$$

As shown in Figure 5, nearby channels have higher correlation coefficients, and the correlation decreases with the increasing of time intervals. Moreover, the decay follows a fluctuating pattern. We then propose to truncate the channel sequence based on the correlation between the channel and $\mathbf{H}_1$. Specifically, if the correlation between $\mathbf{H}_T$ and $\mathbf{H}_1$ is lower than the threshold τ and the correlations between $\mathbf{H}_t$ $(t < T)$ and $\mathbf{H}_1$ are greater than τ, then $\mathbf{H}_T$ is the end of this channel sequence.

4.5 Channel Embedding

To enhance channel representation, we design a dual-path residual embedding architecture:

$$\mathbf{E} = f_{\theta_1}(\mathbf{H}) + g_{\theta_2}(\mathbf{H}), \quad (8)$$

where f_{θ_1} represents the linear embedding layer and g_{θ_2}represents the auxiliary embedding layer trained by masked channel reconstruction.

4.6 Pre-training Tasks

We adopt two pre-training tasks to train CP-GPT: the next channel prediction (NCP) task and the masked channel reconstruction (MCR) task.

4.6.1 Next channel prediction (NCP)

The next channel prediction task includes predicting the next channel matrix and predicting the next channel's angle. Given true next channel $\mathbf{H}_{t+1} \in \mathbb{C}^{N_r \times N_t}$ and the model prediction $\hat{\mathbf{H}}_{t+1} = f_\theta(\mathbf{H}_{t-k:t})$, the loss function of NCP is

$$\begin{aligned}\mathcal{L}_{\text{NCP}} &= \lambda_1 \left\| \hat{\mathbf{H}}_{t+1} - \mathbf{H}_{t+1} \right\|_F^2 \\ &+ \lambda_2 \left(1 - \frac{\langle \text{vec}(\hat{\mathbf{H}}_{t+1}), \text{vec}(\mathbf{H}_{t+1}) \rangle}{\left\| \hat{\mathbf{H}}_{t+1} \right\|_F \|\mathbf{H}_{t+1}\|_F} \right),\end{aligned} \quad (9)$$

where $\lambda_1 + \lambda_2 = 1$.

4.6.2 Masked channel reconstruction (MCR)

The input channel is randomly masked by

$$\tilde{\mathbf{H}} = \mathbf{M} \odot \mathbf{H}, \quad \mathbf{M}_{ij} \sim \text{Bernoulli}(p), \quad (10)$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$ is a binary mask matrix whose elements $\mathbf{M}_{ij}$ are independently sampled from the Bernoulli distribution with parameter p. Then the masked channel is sent to the auxiliary channel embedding layer. Next, the embedding of the masked channel is input into the channel reconstruction network where the output is the reconstructed channel. The loss function of MCR is

$$\mathcal{L}_{\text{MCR}} = \|g_{\theta_2}(\tilde{\mathbf{H}}) - \mathbf{H}\|_F^2. \quad (11)$$

4.6.3 Unified loss function

The loss function encompasses both the NCP loss and the MCR loss is

$$\mathcal{L}_{\text{CP-GPT}} = \beta_1 \mathcal{L}_{\text{NCP}} + \beta_2 \mathcal{L}_{\text{MCR}}, \quad (12)$$

where $\beta_1 + \beta_2 = 1$.

5 Pre-training Dataset

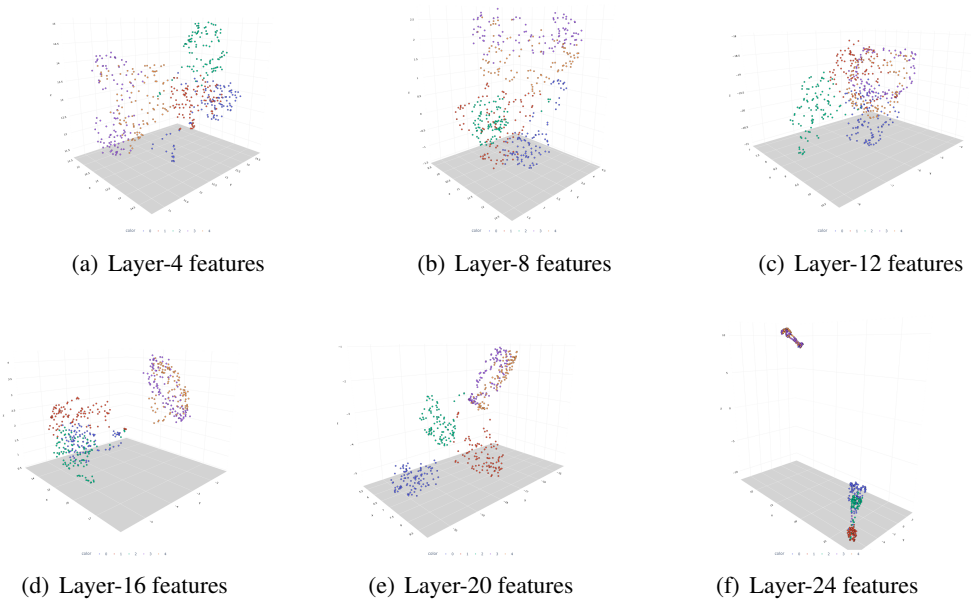
We adopt MATLAB to generate the time-varying CSI data under 3GPP 38.901 standard (3gp, 2019). Specifically, we collect channels with frequency bands ranging from 0.5 GHz to 100 GHz, and terminal movement speeds ranging from 0 km/h to 30 km/h. For each speed and frequency point, we collect five types of channels: CDL-A, CDL-B, CDL-C, CDL-D, and CDL-E. The base station uses an 8x8 dual-polarized planar array, and the user employs a 2×2 planar array. A total of 12M training data is generated.

6 Channel Prediction Benchmark

In this section, we present a comprehensive benchmark designed to evaluate channel prediction models across various dimensions critical for practical communication systems. Specifically, we focus on five key aspects: (i) the few-shot capability of CP-GPT; (ii) the impact of different time slot intervals on prediction accuracy; (iii) the performance under varying numbers of antennas; (iv) the cross-frequency prediction capability; (v) the cross-antenna prediction capability. Each aspect is accompanied by detailed evaluation schemes and datasets to ensure rigorous and reproducible results.

6.1 Few-shot Capability

We construct a few-shot benchmark to evaluate CP-GPT on data from novel scenarios and frequency bands that are not included in the pre-training dataset, supporting both finetuning and evaluation.



(a) Layer-4 features (b) Layer-8 features (c) Layer-12 features

(d) Layer-16 features (e) Layer-20 features (f) Layer-24 features

Figure 6: The clustering results of different layers.

6.2 Impact of Time Slot Intervals

The time slot interval is a crucial factor in channel prediction, as it directly affects the model's ability to capture temporal variations of the channel. To evaluate this effect, we provide datasets with different time slot intervals, i.e., 1 ms, 2 ms, 3 ms, 4 ms, 5 ms, and 6 ms and use normalized mean squared error (NMSE) as the standard evaluation metric.

6.3 Performance under Different Numbers of Antennas

With the advent of 5G and 6G communications systems, antenna arrays have evolved from single-antenna to dozens or even hundreds of antennas. We provide a benchmark with datasets featuring different antenna configurations, ranging from a few antennas to hundreds of antennas, and adopt NMSE as the evaluation metric, supporting both finetuning and evaluation.

6.4 Cross-Frequency Channel Prediction Capability

In wireless communications with Frequency Division Duplex (FDD) mode (Xu et al., 2023), the uplink and downlink signals are transmitted on different frequency bands, which typically requires separate channel estimation for each band. Since the uplink and downlink frequencies are often relatively close, the uplink channel and downlink channel exhibit substantial correlation. Therefore, it is possible to predict the downlink channel based on the uplink channel which can save the pilot overhead. To evaluate this capability, we construct a benchmark for the cross-frequency prediction task, where CP-GPT is finetuned and evaluated on datasets containing channels measured at multiple frequencies.

6.5 Cross-Antenna Channel Prediction Capability

In systems with a large number of antennas, cross-antenna prediction is essential to predict the channel at unobserved antennas based on measurements from observed antennas (Zhang et al., 2021). We construct a cross-antenna prediction benchmark, where CP-GPT is finetuned and evaluated on datasets with different array configurations and antenna sampling densities.

7 Evaluation of CP-GPT

The proposed CP-GPT has 24 decoder layers with 16 attention heads. The hidden dimension is 1024. In total, CP-GPT has 343M parameters. We train CP-GPT for 10 epochs on eight NVIDIA 3090 GPUs using DeepSpeed ZeRO-2, with a learning rate of 4×10^{-6} and a warm-up phase during the first 10% of training steps.

7.1 What Does The CP-GPT Learn

After completing the pre-training, we investigate the internal workings of the CP-GPT to uncover what it has indeed learned. As depicted in Figure

Methods	Time General. ↓	Antenna General. ↓	Few-Shot ↓	Cross-Antenna ↓	Cross-Frequency ↓
Baseline(500)	0.342	0.254	0.238	0.244	0.279
CP-GPT(500)	0.0470	0.0340	0.00572	0.0349	0.0508
Baseline(2k)	0.1117	0.0524	0.0875	0.0845	0.0771
CP-GPT(2k)	**0.00986**	**0.00456**	**0.00133**	**0.00848**	**0.00924**

Table 2: The prediction NMSE of the proposed CP-GPT on the benchmarks. 'Baseline(500 or 2k)' denotes the baseline trained with 500 or 2000 samples; 'CP-GPT(500 or 2k)' represents the proposed CP-GPT fine-tuned with 500 or 2k samples. Traits with _ indicate that the lower the score, the better the performance.

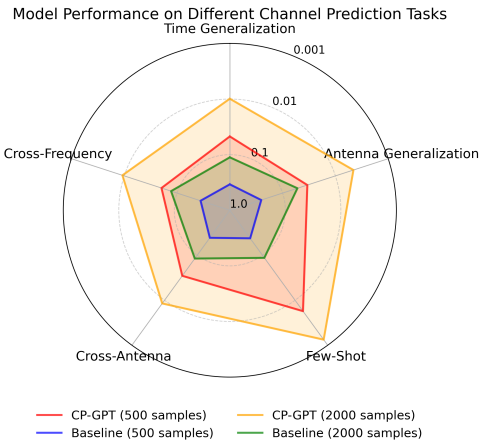


Figure 7: The abilities of the proposed CP-GPT in different downstream tasks.

6, we cluster the output features from several layers of CP-GPT by UMAP (McInnes et al., 2018). Clustering analysis demonstrates a clear separation of the features into two distinct groups in Figure 6(f): one located in the upper left quadrant and the other in the lower right quadrant. The clustering results reveal two groups that predominantly correspond to LOS and NLOS channels, respectively. This finding indicates that the model has autonomously learned to distinguish between these two fundamental channel characteristics, without any explicit prior information about LOS or NLOS properties (Atzeni et al., 2017). The ability to capture such underlying distinctions suggests that CP-GPT has developed a certain level of understanding of channel characteristics, which is beneficial to improve the accuracy and robustness of channel prediction.

7.2 Performance on Benchmark

We test CP-GPT's performance on the benchmark proposed in Section 6. Specifically, we evaluate CP-GPT's performance using 500 and 2000 samples as the adaptation set. We adopt a network with four decoder layers without pre-training as the baseline. As shown in Figure 7 and Table 2, CP-GPT significantly outperforms the baseline on all benchmarks. In particular, on the few-shot channel prediction task with 2000 samples, CP-GPT attains an NMSE as low as 0.00133, which significantly outperforms the baseline with an NMSE of 0.0875. Such superior performance highlights the strong few-shot ability of CP-GPT as well as the effectiveness of the pre-training strategy.

Additional experimental details and methodologies can be found in Appendix B. This comprehensive analysis affirms the robustness and adaptability of CP-GPT in various scenarios, highlighting its potential for broad application in AI-driven channel prediction and beyond.

8 Conclusion

In this work, we propose a novel channel-to-language analogy and designing an LLM-based framework for channel prediction. By treating wireless channels as linguistic units and introducing the concept of an EOCS, we enable the construction of 'channel sentences' that capture temporal correlations in wireless environments. Based on this perspective, we present a unified framework named CP-GPT to train the LLM for channel prediction. We design two dedicated pre-training tasks, NCP and MCR. The NCP task enables CP-GPT to better capture the temporal dynamics of wireless channels, while the MCR task enhances the expressive power of the channel embedding layer by enabling robust channel feature reconstruction. Additionally, we propose a comprehensive benchmark to evaluate LLM-based channel prediction, which encompasses five representative tasks: few-shot channel prediction, channel prediction under different time intervals, channel prediction under different antenna numbers, cross-antenna channel prediction, and cross-frequency channel prediction. Extensive experiments demonstrate the effectiveness and strong generalization ability of CP-GPT.

Limitations

This work is primarily evaluated on synthetic channel data generated according to 3GPP standards, which may not fully reflect real-world propagation conditions due to the absence of commercial network data, as large-scale real-world datasets are often held by mobile network operators such as AT&T, Vodafone Group, and Telstra and are difficult for researchers to access. Nevertheless, we open-source CP-GPT's weights, inference code, and fine-tuning code at https://github.com/linb20/CP-GPT to facilitate validation by researchers and mobile network operators. In addition, we are making efforts to collaborate with mobile network operators to collect additional real-world measurement data.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFE0200700; in part by the National Natural Science Foundation of China under Grant 62325107 and Grant 62261160650; in part by Beijing Natural Science Foundation under Grant L222002: and in part by the Natural Science Foundation of Sichuan Province under Grant 2025YFHZ0022.

References

2019. Study on channel model for frequencies from 0.5 to 100 GHz (Release 16). Technical Report TR 38.901 V16.1.0, 3rd Generation Partnership Project (3GPP).

Italo Atzeni, Jesús Arnau, and Marios Kountouris. 2017. Downlink cellular network analysis with los/nlos propagation and elevated base stations. *IEEE Transactions on Wireless Communications*, 17(1):142–156.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

GSMA. 2025. Mobile World Congress Barcelona 2025. https://www.mwcbarcelona.com/.

Shuaishuai Guo, Yanhu Wang, Jia Ye, Anbang Zhang, Peng Zhang, and Kun Xu. 2025. Semantic importance-aware communications with semantic correction using large language models. *IEEE Transactions on Machine Learning in Communications and Networking*.

Md Arafat Habib, Pedro Enrique Iturria Rivera, Yigit Ozcan, Medhat Elsayed, Majid Bavand, Raimundus Gaigalas, and Melike Erol-Kantarci. 2025. Llm-based intent processing and network optimization using attention-based hierarchical reinforcement learning. In *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

Houda Harkat, Paulo Monteiro, Atilio Gameiro, Fernando Guiomar, and Hasmath Farhana Thariq Ahmed. 2022. A survey on mimo-ofdm systems: Review of recent trends. *Signals*, 3(2):359–395.

Islam Helmy, Pulok Tarafder, and Wooyeol Choi. 2023. Lstm-gru model-based channel prediction for one-bit massive mimo system. *IEEE Transactions on Vehicular Technology*, 72(8):11053–11057.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yang Hong, Jun Wu, and Rosario Morello. 2024. Llm-twin: mini-giant model-driven beyond 5g digital twin networking framework with semantic secure communication and computation. *Scientific Reports*, 14(1):19065.

Ching-Tang Huang, Yu-Chih Huang, Shin-Lin Shieh, and Po-Ning Chen. 2024. Novel prony-based channel prediction methods for time-varying massive mimo channels. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, pages 1–6. IEEE.

Feibo Jiang, Li Dong, Yubo Peng, Kezhi Wang, Kun Yang, Cunhua Pan, and Xiaohu You. 2024. Large ai model empowered multimodal semantic communications. *IEEE Communications Magazine*.

Hao Jiang, Mingyao Cui, Derrick Wing Kwan Ng, and Linglong Dai. 2022. Accurate channel prediction based on transformer: Making mobility negligible. *IEEE Journal on Selected Areas in Communications*, 40(9):2717–2732.

Wei Jiang and Hans D Schotten. 2019. Recurrent neural network-based frequency-domain channel prediction for wideband communications. In *2019 IEEE 89th vehicular technology conference (VTC2019-Spring)*, pages 1–6. IEEE.

Wei Jiang and Hans Dieter Schotten. 2020. Deep learning for fading channel prediction. *IEEE Open Journal of the Communications Society*, 1:320–332.

Benish Sharfeen Khan, Sobia Jangsher, Ashfaq Ahmed, and Arafat Al-Dweik. 2022. Urllc and embb in 5g industrial iot: A survey. *IEEE Open Journal of the Communications Society*, 3:1134–1163.

Hwanjin Kim, Sucheol Kim, Hyeongtaek Lee, Chulhee Jang, Yongyun Choi, and Junil Choi. 2020. Massive mimo channel prediction: Kalman filtering vs. machine learning. *IEEE Transactions on Communications*, 69(1):518–528.

Woongsup Lee and Jeonghun Park. 2024. Llm-empowered resource allocation in wireless communications systems. *arXiv preprint arXiv:2408.02944*.

Qiang Liu, Junsheng Mu, DaRonghui ChenZhang, Yijian Liu, and Tao Hong. 2024. Llm enhanced reconfigurable intelligent surface for energy-efficient and reliable 6g iov. *IEEE Transactions on Vehicular Technology*.

David Löschenbrand, Markus Hofer, and Thomas Zemen. 2023. Spectral efficiency of time-variant massive mimo using wiener prediction. *IEEE Communications Letters*, 27(4):1225–1229.

BMR Manasa and P Venugopal. 2024. A systematic literature review on channel estimation in mimo-ofdm system: Performance analysis and future direction. *Journal of Optical Communications*, 45(3):589–614.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Wanli Ni, Zhijin Qin, Haofeng Sun, Xiaoming Tao, and Zhu Han. 2025. Multi-task semantic communications via large models. *arXiv preprint arXiv:2503.22064*.

Hyeonho Noh, Byonghyo Shim, and Hyun Jong Yang. 2025. Adaptive resource allocation optimization using large language models in dynamic wireless environments. *arXiv preprint arXiv:2502.02287*.

Masayoshi Ozawa, Tomoaki Ohtsuki, Wenjie Jiang, and Yasushi Takatori. 2015. Interference alignment for time-varying channel with channel and weight predictions based on auto regressive model. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. 2024. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 207–222.

Theodore S. Rappaport. 2002. *Wireless Communications: Principles and Practice*, 2nd edition. Prentice Hall.

Oscar Stenhammar, Gabor Fodor, and Carlo Fischione. 2024. A comparison of neural networks for wireless channel prediction. *IEEE Wireless Communications*, 31(3):235–241.

Geng Sun, Yixian Wang, Dusit Niyato, Jiacheng Wang, Xinying Wang, H Vincent Poor, and Khaled B Letaief. 2024. Large language model (llm)-enabled graphs in dynamic networking. *IEEE Network*.

Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. 2024. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 36.

Cheng-Xiang Wang, Junling Li, Jie Huang, Chen Huang, Zhaoyang Zhang, Ying Liu, Shidong Zhou, Yunfei Chen, Xiaohu You, Xiqi Gao, et al. 2025. Modeling, capacity studies, antenna and system designs for 6g/b6g 3d continuous-space radio channels enabled by electromagnetic information theory. *IEEE Communications Surveys & Tutorials*.

Xianling Wang, Yi Shi, Wenbo Xin, Tianci Wang, Guangli Yang, and Zhiyuan Jiang. 2023. Channel prediction with time-varying doppler spectrum in high-mobility scenarios: A polynomial fourier transform based approach and field measurements. *IEEE Transactions on Wireless Communications*, 22(11):7116–7129.

Huiqiang Xie, Zhijin Qin, Xiaoming Tao, and Zhu Han. 2024. Towards intelligent communications: Large model empowered semantic communications. *arXiv preprint arXiv:2402.13073*.

Huixin Xu, Jianhua Zhang, Pan Tang, Lei Tian, Qixing Wang, and Guangyi Liu. 2023. An empirical study on channel reciprocity in tdd and fdd systems. *IEEE Open Journal of Vehicular Technology*, 5:108–124.

Poonam Yadav, Sandeep Kumar, and Rajesh Kumar. 2021. A comprehensive survey of physical layer security over fading channels: Classifications, applications, and challenges. *Transactions on Emerging Telecommunications Technologies*, 32(9):e4270.

Wanting Yang, Zehui Xiong, Shiwen Mao, Tony QS Quek, Ping Zhang, Merouane Debbah, and Rahim Tafazolli. 2024. Rethinking generative semantic communication for multi-user systems with multi-modal llm. *arXiv preprint arXiv:2408.08765*.

Haifan Yin, Haiquan Wang, Yingzhuang Liu, and David Gesbert. 2020. Dealing with the mobility problem of massive mimo using extended prony's method. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Yong Zeng, Junting Chen, Jie Xu, Di Wu, Xiaoli Xu, Shi Jin, Xiqi Gao, David Gesbert, Shuguang Cui, and Rui Zhang. 2024. A tutorial on environment-aware communications via channel knowledge map for 6g. *IEEE communications surveys & tutorials*, 26(3):1478–1519.

Shun Zhang, Yushan Liu, Feifei Gao, Chengwen Xing, Jianping An, and Octavia A Dobre. 2021. Deep learning based channel extrapolation for large-scale antenna systems: Opportunities, challenges and solutions. *IEEE Wireless Communications*, 28(6):160–167.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2024a. A comprehensive survey on

Tasks	Z-S ↓	F-S(0.5k) ↓	F-S(2k) ↓
NCP only	0.4704	0.00735	0.00364
NCP+MCR	**0.1644**	**0.00572**	**0.00133**

Table 3: Performance comparison of pre-training tasks. "Z-S" denotes zero-shot; "F-S(0.5k)" and "F-S(2k)" denote few-shot with 0.5k and 2k samples, respectively.

pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, pages 1–65.

Hao Zhou, Chengming Hu, Dun Yuan, Ye Yuan, Di Wu, Xue Liu, and Charlie Zhang. 2024b. Large language model (llm)-enabled in-context learning for wireless network optimization: A case study of power control. *arXiv preprint arXiv:2408.00214*.

Tao Zhou, Haitong Zhang, Bo Ai, Chen Xue, and Liu Liu. 2022. Deep-learning-based spatial–temporal channel prediction for smart high-speed railway communication networks. *IEEE Transactions on Wireless Communications*, 21(7):5333–5345.

A Channel Model Illustration

For a system with N_t transmit antennas and N_r receive antennas, the time-domain channel matrix $\mathbf{H}_n$ at the n-th sample interval is modeled as:

$$\mathbf{H}_n = \sqrt{N_r N_t} \sum_{l=0}^{L_c-1} \alpha_l g(nT - \tau_l) \times \mathbf{a}_r(\phi_{r,l}, \theta_{r,l}) \mathbf{a}_t^*(\phi_{t,l}, \theta_{t,l}), \quad (13)$$

where L_c represents the total number of multipath components, α_l represents the complex gain of the l-th path, τ_l represents the propagation delay associated with the l-th path, $g(\cdot)$ represents the pulse-shaping filter evaluated at time $nT - \tau_l$, $\phi_{r,l}$ represents the azimuth angle of arrival (AoA) of the l-th path, $\theta_{r,l}$ represent the elevation AoA of the l-th path, $\phi_{t,l}$ represents the azimuth angle of departure (AoD) of the l-th path, $\theta_{t,l}$ represents the elevation AoD of the l-th path, $\mathbf{a}_r(\phi_{r,l}, \theta_{r,l})$ represents the receive steering vector, and $\mathbf{a}_t(\phi_{t,l}, \theta_{t,l})$ represents the transmit steering vector. For a uniform planar array (UPA) with N_x and N_y elements along the x and y axes, the steering vector can be calculated by

$$\mathbf{a}_{\text{UPA}}(\phi, \theta) = \mathbf{a}_x(\phi, \theta) \otimes \mathbf{a}_y(\phi, \theta), \quad (14)$$

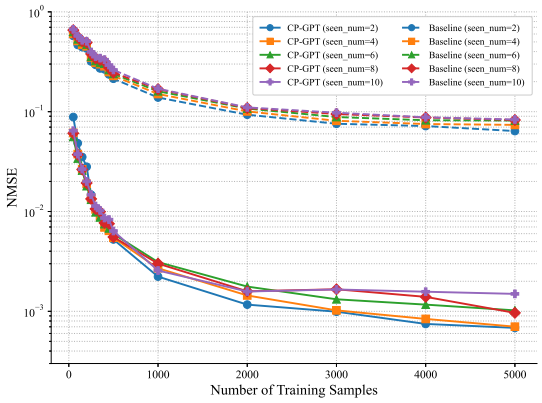


Figure 8: Performance of the proposed CP-GPT in the few-shot benchmark.

$$\mathbf{a}_x(\phi, \theta) = \frac{1}{\sqrt{N_x}} \begin{bmatrix} 1 \\ e^{j\frac{2\pi d_x \sin\theta\cos\phi}{\lambda}} \\ \vdots \\ e^{j\frac{2\pi d_x (N_x-1)\sin\theta\cos\phi}{\lambda}} \end{bmatrix}, \quad (15)$$

$$\mathbf{a}_y(\phi, \theta) = \frac{1}{\sqrt{N_y}} \begin{bmatrix} 1 \\ e^{j\frac{2\pi d_y \sin\theta\sin\phi}{\lambda}} \\ \vdots \\ e^{j\frac{2\pi d_y (N_y-1)\sin\theta\sin\phi}{\lambda}} \end{bmatrix}, \quad (16)$$

where λ represents the wavelength, d_x and d_y represent the antenna spacings along the x and y axes.

B Ablation Studies and Details of Channel Prediction Benchmark

B.1 Ablation Studies on MCR

We conducted comparative experiments between CP-GPT trained with "NCP only" and CP-GPT trained with "NCP + MCR". The results are summarized in Table 3. It can be seen that introducing MCR consistently reduces the NMSE in both zero-shot and few-shot settings.

B.2 Few-Shot Performance

Few-shot evaluation is conducted on data that is strictly excluded from the pre-training process to rigorously assess the generalization ability of CP-GPT. Specifically, frequency points used for pre-training are $\{0.5, 1.0, 1.5, \ldots, 100\}$ GHz, whereas those for testing are $\{0.75, 1.25, 1.75, \ldots\}$ GHz, ensuring no overlap between the two sets. The dimension of input antennas is set to 128.

The fine-tuning results of CP-GPT are shown in Figure 8. It can be observed that CP-GPT achieves a prediction accuracy of NMSE 0.01 with only 350

Group	SFT Samples	CP-GPT					Baseline				
		2ms	3ms	4ms	5ms	6ms	2ms	3ms	4ms	5ms	6ms
Small	100	0.0742	0.1035	0.1445	0.2275	0.2852	0.6055	0.6250	0.6953	0.7930	0.8516
	150	0.0669	0.0913	0.1260	0.2012	0.2500	0.5508	0.5664	0.6602	0.7422	0.7773
	200	0.0603	0.0801	0.1045	0.1592	0.1768	0.5195	0.5391	0.6133	0.6875	0.7188
	250	0.0493	0.0654	0.0830	0.1279	0.1377	0.4141	0.4434	0.5078	0.5508	0.6094
	300	0.0420	0.0554	0.0713	0.1060	0.1138	0.3789	0.4004	0.4531	0.5078	0.5469
	350	0.0393	0.0520	0.0669	0.0991	0.1035	0.3555	0.3867	0.4453	0.4883	0.5234
	400	0.0356	0.0457	0.0566	0.0806	0.0859	0.3516	0.3809	0.4258	0.4648	0.5078
	450	0.0334	0.0437	0.0544	0.0742	0.0786	0.3066	0.3438	0.4023	0.4336	0.4492
	500	0.0283	0.0356	0.0442	0.0601	0.0669	0.2734	0.3027	0.3457	0.3809	0.4063
Medium	1000	0.0159	0.0188	0.0210	0.0269	0.0308	0.1777	0.1953	0.2188	0.2324	0.2422
	2000	0.0099	0.0111	0.0121	0.0144	0.0167	0.1133	0.1250	0.1348	0.1426	0.1592
	3000	0.0080	0.0085	0.0095	0.0107	0.0126	0.0947	0.1016	0.1123	0.1191	0.1309
	4000	0.0070	0.0073	0.0080	0.0092	0.0106	0.0854	0.0977	0.1069	0.1094	0.1187
	5000	0.0064	0.0067	0.0070	0.0082	0.0093	0.0845	0.0913	0.0972	0.1040	0.1167

Table 4: The performance of CP-GPT under different time intervals. Here, $t$ms denotes the time interval between input channels.

Group	SFT Samples	$N_{\text{seen}} = 2$		$N_{\text{seen}} = 4$		$N_{\text{seen}} = 6$		$N_{\text{seen}} = 8$		$N_{\text{seen}} = 10$	
		CP-GPT	Base.	CP-GPT	Base.	CP-GPT	Base.	CP-GPT	Base.	CP-GPT	Base.
Small	100	0.155 273	0.566 406	0.121 094	0.574 219	0.132 812	0.589 844	0.128 906	0.589 844	0.125 977	0.593 750
	150	0.140 625	0.558 594	0.113 281	0.554 688	0.126 953	0.574 219	0.122 559	0.566 406	0.118 652	0.570 312
	200	0.125 000	0.523 438	0.100 586	0.523 438	0.110 352	0.546 875	0.104 980	0.539 062	0.100 586	0.550 781
	250	0.101 562	0.402 344	0.083 496	0.410 156	0.090 820	0.417 969	0.084 961	0.414 062	0.082 520	0.412 109
	300	0.091 309	0.371 094	0.075 684	0.380 859	0.082 520	0.396 484	0.076 172	0.396 484	0.074 219	0.394 531
	350	0.080 566	0.355 469	0.065 918	0.367 188	0.069 824	0.376 953	0.064 941	0.369 141	0.063 965	0.371 094
	400	0.074 219	0.349 609	0.060 303	0.355 469	0.063 965	0.361 328	0.059 814	0.363 281	0.057 861	0.376 953
	450	0.063 477	0.296 875	0.054 932	0.298 828	0.058 838	0.306 641	0.055 420	0.306 641	0.053 223	0.300 781
	500	0.056 396	0.273 438	0.049 072	0.277 344	0.052 246	0.281 250	0.048 828	0.279 297	0.047 363	0.285 156
Medium	1000	0.027 222	0.157 227	0.023 926	0.163 086	0.023 926	0.166 016	0.023 804	0.166 992	0.022 827	0.171 875
	2000	0.013 672	0.090 820	0.012 390	0.095 215	0.012 329	0.097 168	0.012 024	0.096 191	0.011 841	0.098 633
	3000	0.010 010	0.071 777	0.009 216	0.075 195	0.009 216	0.078 125	0.008 850	0.079 590	0.008 911	0.081 055
	4000	0.008 118	0.064 941	0.007 599	0.065 430	0.007 660	0.066 895	0.007 446	0.068 848	0.007 599	0.071 777
	5000	0.006 989	0.057 129	0.006 653	0.061 279	0.006 744	0.062 988	0.006 622	0.062 988	0.006 805	0.065 430

Table 5: The performance of CP-GPT for cross-frequency channel prediction. Here, N_{seen} represents the input channel sequence length.

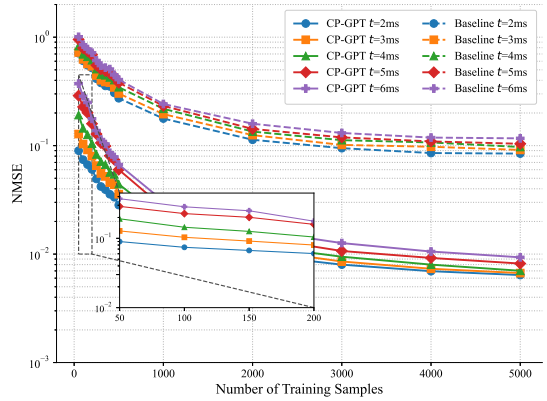


Figure 9: Performance of the proposed CP-GPT with different time intervals.

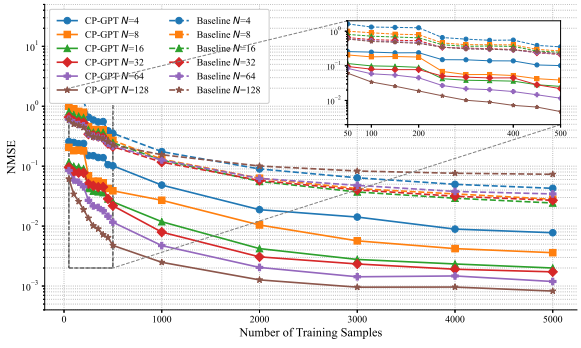


Figure 10: Performance of the proposed CP-GPT with different antenna configurations.

fine-tuning samples. In wireless communications systems, the transmission bandwidth is typically divided into multiple narrow frequency bands called subcarriers. Notably, if 512 subcarriers are used for signal transmission, then the base station can obtain 512 channel samples from a single user within one time slot, which implies that obtaining 350 channel samples within a cell is feasible for the communications systems.

Furthermore, CP-GPT demonstrates a significant improvement of two orders of magnitude in performance compared to the baseline that is trained from scratch. This highlights the effectiveness of CP-GPT's pre-training and fine-tuning strategy in few-shot applications.

Group	SFT Samples	CP-GPT				Baseline			
		$N_a = 8$	$N_a = 16$	$N_a = 32$	$N_a = 64$	$N_a = 8$	$N_a = 16$	$N_a = 32$	$N_a = 64$
Small	100	0.6680	0.2578	0.0688	0.0486	0.8398	0.6172	0.5625	0.5586
	150	0.6523	0.2432	0.0649	0.0449	0.8086	0.5938	0.5391	0.5273
	200	0.6250	0.2324	0.0605	0.0413	0.7812	0.5859	0.5195	0.5234
	250	0.5742	0.1934	0.0508	0.0369	0.6719	0.4375	0.3711	0.3672
	300	0.5430	0.1758	0.0464	0.0320	0.6289	0.4063	0.3320	0.3379
	350	0.5234	0.1689	0.0449	0.0308	0.6250	0.3926	0.3242	0.3223
	400	0.5039	0.1572	0.0427	0.0288	0.6055	0.3770	0.3184	0.3184
	450	0.4629	0.1367	0.0393	0.0281	0.5234	0.3223	0.2617	0.2598
	500	0.4434	0.1187	0.0349	0.0242	0.4844	0.2949	0.2441	0.2471
Medium	1000	0.3125	0.0605	0.0217	0.0153	0.3105	0.1826	0.1563	0.1553
	2000	0.1660	0.0306	0.0119	0.0092	0.1738	0.1201	0.1021	0.1016
	3000	0.1084	0.0217	0.0085	0.0070	0.1318	0.0957	0.0845	0.0835
	4000	0.0796	0.0167	0.0068	0.0059	0.1152	0.0879	0.0767	0.0771
	5000	0.0623	0.0139	0.0059	0.0052	0.1021	0.0801	0.0708	0.0688

Table 6: The performance of CP-GPT for cross-antenna channel prediction. Here, N_a denotes the input antenna dimension, while the output dimension is fixed at 128 antennas.

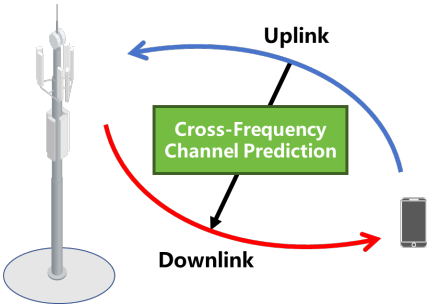


Figure 11: Cross-frequency channel prediction description.

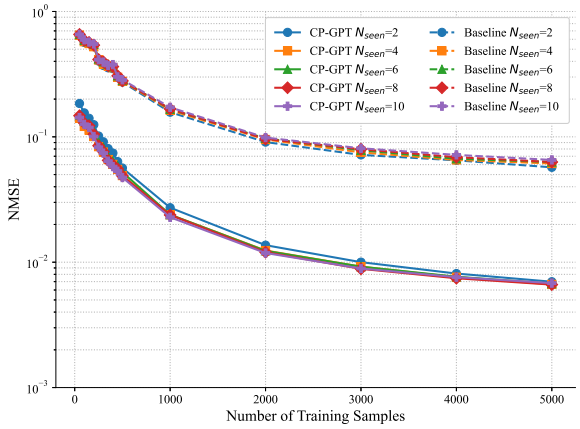


Figure 12: Performance of the proposed CP-GPT in the cross-frequency channel prediction task.

B.3 Generalization Across Different Time Intervals

To further investigate the generalization capability of CP-GPT across different time intervals, we construct datasets with time intervals ranging from 2 ms to 6 ms and fine-tuned CP-GPT on these datasets. As shown in Figure 9 and Table 4, CP-GPT demonstrates strong generalization over sequential inputs with different time intervals, achieving NMSE on the order of 1×10^{-2}. Notably, CP-GPT outperforms baseline by an order of magnitude in channel prediction accuracy. This highlights the robustness of CP-GPT in handling temporal data with diverse time intervals.

B.4 Performance under Different Numbers of Antennas

The number of antennas can differ across base stations. We then evaluate CP-GPT's ability to adapt to different antenna configurations. We conduct experiments with antenna numbers 128, 64, 32, 16, 8, and 4, respectively. The results are illustrated in Figure 10. It can be seen that CP-GPT achieves an NMSE of 0.01 with 16, 32, and 64 antennas using only 1000 training samples. Compared to the baseline, CP-GPT demonstrates a significant improvement of one order of magnitude in prediction accuracy. This further validates the robustness and adaptability of CP-GPT in diverse antenna configurations with limited training data.

B.5 Cross-Frequency Channel Prediction Capability

There are two typical communication modes in wireless communication systems: time division duplex (TDD) and frequency division duplex (FDD). In systems with TDD mode, channel reciprocity allows the downlink channel to be inferred from the uplink channel. Hence, channel estimation needs to be performed only for the uplink. In contrast,

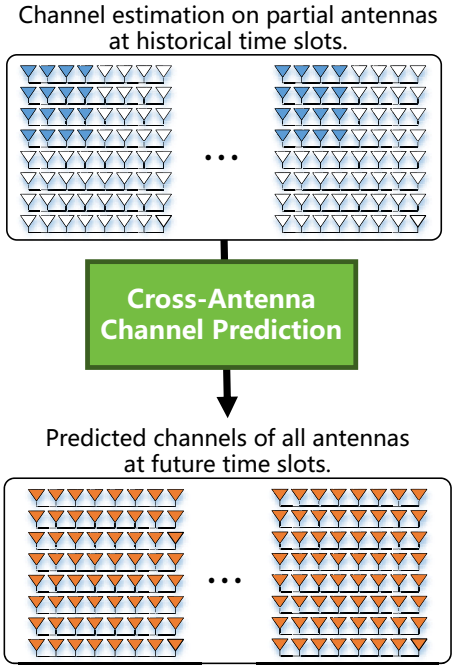


Figure 13: Cross-antenna channel prediction description.

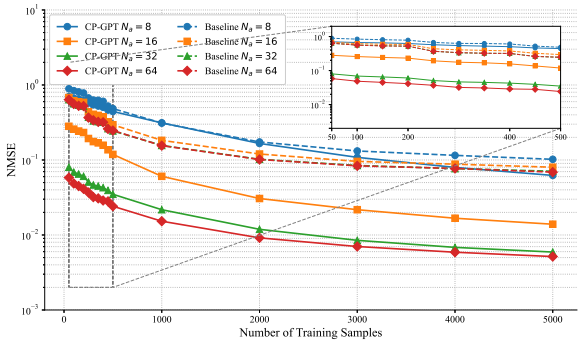


Figure 14: Performance of the proposed CP-GPT in the cross-antenna channel prediction task.

in systems with FDD mode, the uplink and downlink channels operate in different frequency bands, which breaks the channel reciprocity assumption. As a result, both the uplink and downlink channels must be estimated separately, which leads to increased pilot overhead. Therefore, cross-frequency channel prediction as shown in Figure 11 is a crucial technique in FDD mode, as it enables the prediction of downlink channels from uplink channels to save the pilots.

To evaluate CP-GPT's capability in cross-frequency channel prediction, we conduct fine-tuning datasets containing samples from multiple frequency bands. We then fine-tune CP-GPT on the cross-frequency dataset. As shown in Figure 12 and Table 5, CP-GPT achieves an NMSE on the order of 1×10^{-2} and outperforms the baseline by about an order of magnitude.

B.6 Cross-Antenna Channel Prediction Capability

As the size of antenna arrays increases, the pilot overhead of channel acquisition grows proportionally. Cross-antenna channel prediction, which involves predicting the channels of partial antennas from those of other partial antennas as shown in Figure 13, has emerged as an effective method to reduce the pilot overhead for channel acquisition.

Figure 14 and Table 6 demonstrate CP-GPT's performance in cross-antenna prediction. With only 500 samples, CP-GPT achieves an NMSE of 1×10^{-2} when predicting the channels of 128 antennas from the other 32 antennas. Furthermore, when predicting the channels of all 128 antennas from only 16, 32, and 64 antennas, CP-GPT outperforms baseline methods by approximately an order of magnitude.