# Can LLMs Compress Wireless Signals? Physics-Informed Prompting for Training-Free Signal Compression

**Yiwei Wang[1], Lixiang Lian[1†], Sijie Ji[2†]**

[1]School of Information Science and Technology, ShanghaiTech University, China
[2]Division of Engineering and Applied Science, California Institute of Technology, USA
wangyw12024@shanghaitech.edu.cn, lianlx@shanghaitech.edu.cn, sijieji@caltech.edu

## Abstract

Wireless signals are a unique form of time-series data that encode rich physical characteristics of the propagation environment. As large volumes of signals are continuously generated in wireless sensing systems, efficient compression is essential for scalable transmission under limited system resources. However, wireless signals exhibit strong variability across environments and time, making it difficult to design universal compression schemes that generalize across domains while preserving the physical fidelity required for accurate reconstruction and sensing. Meanwhile, recent advances in large language models (LLMs) have demonstrated remarkable reasoning and adaptation capabilities across diverse modalities and domains. This raises an intriguing question: **can LLMs compress wireless signals in a training-free manner while preserving physical fidelity?** To the best of our knowledge, this problem remains unexplored. To answer this question, we propose a physics-informed prompting framework for training-free wireless signal compression. The framework incorporates environmental context and scenario-specific knowledge as physical priors in the prompt. We further introduce a dictionary module with self-learned patterns to stabilize inference and a key prompt module to align the semantics between the LLMs used for compression and reconstruction. Extensive experiments demonstrate that the proposed framework achieves high-fidelity reconstruction under cross-domain settings, highlighting the promise of physics-informed prompting as a new paradigm for processing and compressing low-level physical signals.[1]

## 1 Introduction

Wireless signals are becoming a fundamental substrate for both modern communication (Wen et al., 2018; Yuan et al., 2025; Tian and Lian, 2024; Liu et al., 2024; Guo et al., 2020) and emerging sensing systems (Wang et al., 2024; Li et al., 2024). Beyond carrying information for connectivity, they also encode rich physical interactions with surrounding environments, enabling a wide range of applications such as human activity recognition (HAR) (Ji et al., 2024; Haque et al., 2023), gesture detection (Hu et al., 2023), and indoor localization (Wu et al., 2023). This dual role makes wireless signals uniquely valuable: they are not merely generic time-series measurements, but low-level physical signals whose variations are shaped by propagation, reflection, scattering, and multipath effects. Preserving these fine-grained physical characteristics is essential for downstream sensing tasks. At the same time, the growing deployment of wireless sensing in edge and resource-constrained platforms has created an urgent need for efficient signal compression. Continuous wireless measurements often generate large volumes of data, while practical systems are limited by bandwidth, storage, latency, and energy budgets. Efficient compression is therefore critical for scalable transmission and deployment. However, compressing wireless signals is fundamentally challenging. Unlike conventional signals whose statistical structure is relatively stable, wireless signals exhibit strong variability across environments, devices, human activities, and deployment conditions. Their representations are highly sensitive to physical context, making it difficult to design universal compression schemes that generalize across scenarios while still preserving the physical fidelity required for accurate reconstruction and sensing.

Recent advances in large language models (LLMs) (Yang et al., 2025; Aydin et al., 2025; DeepSeek-AI et al., 2025; Team et al., 2025) have demonstrated strong reasoning and adaptation abilities across diverse modalities, especially in zero-shot settings where explicit task-specific training is

†Corresponding authors.

[1]Our code is available at https://github.com/ywWang-coder/LLM-Signal-Compression



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 18553–18573
October 24-29, 2026 ©2026 Association for Computational Linguistics

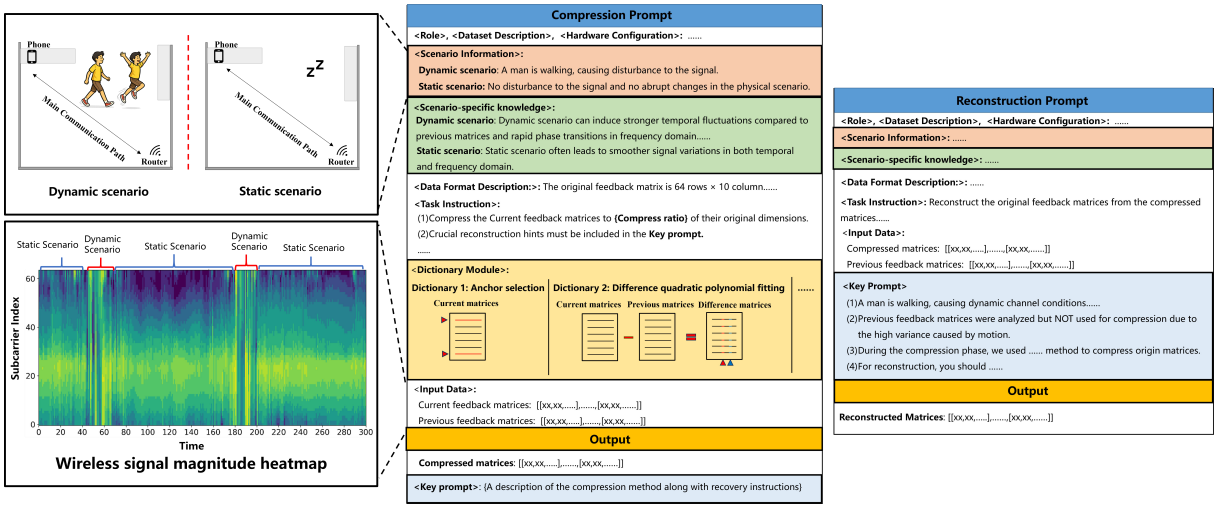



Figure 1: Overview of physics-informed prompt. The complete version of the prompt is provided in Appendix D.

unavailable. This raises a key question: **can LLMs enable training-free wireless signal compression while preserving physical signal fidelity?** In recent studies, prompt-guided approaches have been developed in (Ren et al., 2025; An et al., 2025) to leverage the physical features of wireless data for sensing tasks. A fine-tuned LLM was employed in Cui et al. (2025) for channel state information compression and reconstruction. Prompt engineering was leveraged in Yang et al. (2024) for transportation data compression and reconstruction. However, to the best of our knowledge, the training-free wireless signal compression capability of LLMs has not yet been explored.

Enabling LLMs to perform training-free **W**ireless signal **C**ompression and **R**econstruction (WCR) poses several fundamental challenges. **First**, there exists significant modality mismatch between LLMs and wireless signals. LLMs are designed for discrete token manipulation and semantic reasoning within natural language contexts. Unlike natural language inputs, raw wireless signals are abstract numerical sequences whose underlying physical semantics are difficult to infer without explicit priors. **Second**, WCR requires recovering physically consistent signals from compressed representations, where part of the information is inevitably lost during compression. Unlike language generation tasks where multiple plausible outputs may be acceptable, wireless signals must preserve strict structural and physical constraints imposed by propagation dynamics. **Third**, LLM generation may exhibit stochastic variability, which can introduce instability when reconstructing precise physical signals, making high-fidelity recovery more difficult than typical language generation tasks. **Finally**, in a training-free setting, the compressor and reconstructor must maintain consistent interpretation of the compressed representation without joint training. This becomes particularly challenging when heterogeneous compressor and reconstructor LLMs are used, as their distinct knowledge bases may cause semantic misalignment.

Motivated by these challenges, as shown in Figure 1, we propose physics-informed prompting, a training-free framework for exploring the capability of LLMs in WCR. The proposed prompting scheme consists of three main components.

**1. Physics-Informed Priors**: To facilitate LLM understanding of wireless signals and preserve physical consistency during reconstruction, we inject physics-informed priors into the prompt. These priors provide contextual guidance that helps the LLM interpret the semantic meaning and structural patterns of wireless signals. Specifically, the priors include two parts: (i) *scenario information*, describing the propagation environment (e.g., static or dynamic scenarios), and (ii) *scenario-specific knowledge*, which encodes analytical insights and known correspondences between signal features and sensing scenarios.

**2. Dictionary Module**: To mitigate the stochastic instability of LLM outputs, we introduce a dictionary module that contains a small set of self-learned reference patterns extracted from high-fidelity WCR trajectories generated by the LLMs. These exemplars serve as stabilizing anchors that constrain the generation space while preserving the flexibility to produce adaptive strategies.

**3. Key-Prompt Module**: To ensure semantic align-

ment between the compressor and reconstructor LLMs, we propose a key prompt module generated during compression. This module records essential compression logic and structural cues, enabling the reconstructor to correctly interpret the compressed representation and recover missing information.

Overall, we make three main contributions: 1) We propose a physics-informed prompt strategy, effectively guiding LLMs to apply domain-specific knowledge to WCR, thereby achieving high-fidelity reconstruction. 2) We design a dictionary module to improve inference stability and a key prompt module to align the knowledge contexts between compressor and reconstructor LLMs. 3) Our extensive experiments and comprehensive ablation studies on wireless datasets demonstrate that LLMs have the ability for WCR, paving the way for future research on adopting LLMs for cyber-physical systems.

## 2 LLMs for Wireless Signals

Recent advancements in LLMs (Yang et al., 2025; Aydin et al., 2025; DeepSeek-AI et al., 2025; Team et al., 2025) have demonstrated their ability to training-free mimic human reasoning at a high level and their exceptional generalization abilities. These attributes are pivotal for enabling LLMs to derive deep semantic insights from wireless signals, effectively bridging the gap between abstract numerical sequences and fundamental physical principles. For example, Wi-chat (Ren et al., 2025) integrated physical principles of Wi-Fi sensing into prompts, guiding LLMs to interpret wireless signals for HAR. IoT-LLM (An et al., 2025) employed IoT-oriented retrieval-augmented generation (RAG) to expand the model's domain knowledge, utilizing Chain-of-Thought (CoT) prompting to activate physical reasoning capabilities for complex sensing tasks. These studies show that LLMs can capture certain semantic patterns from physical signals. However, existing sensing tasks primarily rely on extracting condensed features for downstream inference. In contrast, WCR requires generating physically consistent signals governed by complex propagation dynamics, which demands a deeper understanding of the global structure of wireless signals, making WCR fundamentally more challenging.

LLMs have been adopted for WCR. For example, Cui et al. (2025) adopted GPT-2 as a backbone, augmenting it with additional neural layers and fine-tuning on specific datasets to achieve WCR. However, such a fine-tuning method entails a significant dependency on large-scale datasets and imposes substantial computational overhead. Yang et al. (2024) has proposed a zero-shot paradigm for transportation data compression and reconstruction, but it does not account for the propagation physics of wireless signals. The use of prompt engineering to explicitly leverage the physical semantics of wireless signals for compression and reconstruction remains largely unexplored. To address this gap, we introduce physics-informed prompting to investigate the potential of LLMs for WCR.

## 3 Physics-Informed Prompt

### 3.1 Processing Wireless Signals with LLMs

Although LLMs are originally designed for language modeling, their sequence modeling capability is not restricted to natural-language tokens. Wireless signals are discrete sequences with strong temporal, spatial, and frequency-domain correlations governed by physical propagation. Similar to natural language, these sequences contain structured dependencies that can be reasoned about. LLM is not asked to infer physical meaning from numbers alone. We provide physics-informed priors (e.g., propagation conditions and domain knowledge) that connect signal patterns with their underlying physical causes. The LLM then reasons over the relationships among consecutive wireless signals under this physical context, rather than treating them as isolated numerical values. Therefore, our contribution is not that LLMs naturally understand wireless signals, but that appropriate physical guidance enables general-purpose LLMs to process structured physical sequences without task-specific training.

### 3.2 Problem Setting

In this paper, we formulate the problem of WCR for continuous wireless signal streams. Let $X_t$ denote the wireless signal at time frame $t$. Instead of compressing $X_t$ independently, the compressor leverages one or multiple historical frames $\{X_{t-k}\}_{k=1}^{K}$ as references to encode the current signal $X_t$ into a compressed representation $Z_t$. Compression ratio $\alpha$ is equal to $dim(Z_t)/dim(X_t)$. During reconstruction, the reconstructor utilizes the previously recovered wireless signals from earlier time frames $\{\hat{X}_{t-k}\}_{k=1}^{K}$ as references to restore the current frame $\hat{X}_t$ from $Z_t$. The objective is to mini-

mize reconstruction distortion between $X_t$ and $\hat{X}_t$.

**Metrics.** To evaluate the fidelity of reconstruction, we adopt the normalized mean square error (NMSE), a standard metric widely utilized in the signal processing domain. The NMSE is formulated as:

$$\text{NMSE} = \frac{\|X_t - \hat{X}_t\|_F^2}{\|X_t\|_F^2}. \quad (1)$$

A lower NMSE value indicates a smaller discrepancy between the original and reconstructed data. We express the NMSE in decibels (dB) throughout our empirical evaluations. The conversion to the logarithmic scale is defined as:

$$\text{NMSE (dB)} = 10 \cdot \log_{10}(\text{NMSE}). \quad (2)$$

### 3.3 Compression

In the compression, we construct an augmented prompt to guide the LLM in learning an optimal compression strategy. Beyond the standard prompt components (role declaration, hardware configuration, dataset description, and task instructions), the proposed structured prompt further introduces physics-informed priors $\mathcal{PI}$ comprising scenario information $\mathcal{S}$ and scenario-specific knowledge $\mathcal{P}$, together with a dictionary module $\mathcal{D}$. After compression, the compressor LLM generates a key prompt $\mathcal{K}$ that is shared with the reconstructor.

**Scenario Information.** This component describes the physical propagation environment of wireless signals. For example, a "*dynamic scenario*" indicates "*human motion*" activity along the dominant propagation path, while a "*static scenario*" refers to an environment without disturbances along the main path. Such contextual information helps the LLM interpret signal variations under different physical conditions.

**Scenario-Specific Knowledge.** This component encodes prior knowledge about the correspondence between scenario conditions and wireless signal variation patterns. For instance, the knowledge priors may describe that "*dynamic scenario can induce stronger temporal fluctuations and rapid phase transitions in frequency domain*", or "*static scenario often leads to smoother signal variations in both temporal and frequency domain*". These knowledge priors guides the LLM to better capture the structural patterns in wireless signals during compression, enabling more physically consistent recovery.

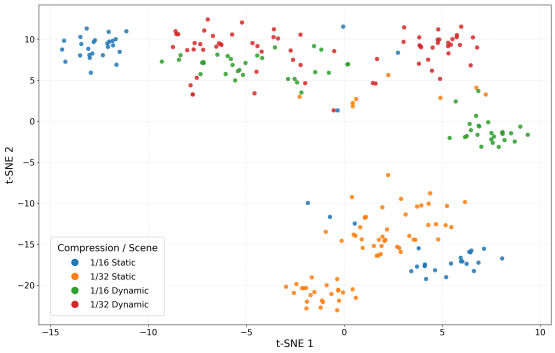



Figure 2: t-SNE visualization of key-prompt embeddings encoded by BGE-M3.

**Dictionary.** This module maintains a set of self-learned reference compression patterns derived from high-fidelity WCR trajectories generated by LLMs. For example, the *"anchor selection"* method selects a few representative rows as anchors to assist the WCR process, while the *"difference quadratic polynomial fitting"* method applies quadratic polynomial fitting to the differences between consecutive wireless signal samples. Unlike conventional few-shot prompting that relies on dataset-specific input-output instances, this dictionary serves as a reusable, prompt-level knowledge base of high-level strategies. This enables direct application to unseen data without requiring target-distribution examples or model fine-tuning.

**Key Prompt.** This module is generated by the compressor LLM to guide the reconstructor LLM in interpreting the compression logic. It records essential information about the compression process, including whether historical frames are utilized, the compression strategy applied, and structural cues that facilitate reconstruction.

Thus, the key prompt is not instance-specific. Instead, it encodes coarse-grained environmental characteristics and compression strategies, which are shared across multiple samples collected under similar conditions. To empirically examine this property, we vectorize the key prompts using the semantic encoder BGE-M3 and visualize their distributions with t-SNE. As shown in Figure 2, signals collected under similar physical environments and compression ratios typically follow similar compression patterns, and their corresponding key prompts tend to concentrate into several dominant clusters.

The key prompt can be indexed from a finite prompt set, where each entry corresponds to an

environment–compression, thereby significantly reducing the transmission cost. We further analyze this compression method in Section 5.3.

Guided by these rich physical and empirical priors, the compressor $\mathcal{C}_{LLM}$ generates a compressed representation and a key prompt:

$$[Z_t, \mathcal{K}] = \mathcal{C}_{\text{LLM}}\left(X_t \mid \{X_{t-k}\}_{k=1}^{K}, \mathcal{S}, \mathcal{P}, \mathcal{D}, \alpha\right). \tag{3}$$

### 3.4 Reconstruction

During the reconstruction phase, the LLM leverages its inherent reasoning and generative capabilities to decode the compressed wireless signal $Z_t$ under the assistance of key prompt $\mathcal{K}$. Consistent with the compression stage, we also embed rich physical features into the prompt of reconstructor to guide LLM to comprehend the physical signals, thereby maximizing its generative capacity to achieve high-fidelity reconstruction. Formally, the reconstructor $\mathcal{R}_{LLM}$ processes the compressed representation $Z_t$ to output the recovered current frame $\hat{X}_t$ as follows:

$$\hat{X}_t = \mathcal{R}_{LLM}(Z_t|\{X_{t-k}\}_{k=1}^{K}, \mathcal{K}, \mathcal{S}, \mathcal{P}, \alpha). \tag{4}$$

## 4 Experimental Setup

In this section, we describe the details of datasets, baseline methods and the implementation settings. More details are included in Appendix A.

### 4.1 Datasets

We evaluate the proposed physics-informed prompt using our self-collected Wi-CBR dataset, which comprises wireless signal frames recorded over two days within the same indoor environment. Furthermore, we evaluate the generalization of our approach across datasets from diverse real-world scenarios including outdoor settings, varying communication bandwidths, and distinct sampling strategies as well as on the public Wi-BFI indoor dataset. Given the sensitivity of wireless signal to even minute physical perturbations, data collected from the identical scenario across different periods exhibits substantial structural differences. For each time period, the dataset include two scenarios: a dynamic scenario, in which a person walks along the main communication path and introduces interference, and a static scenario, in which no human interference is present. These scenarios act as robust abstractions of physical propagation conditions, allowing us to focus on generalizable, physics-informed priors rather than exhaustively modeling system-specific signal parameters, which are more dependent on specific devices. More evaluation and discussions regarding diverse wireless signal characteristics are provided in Appendix B.1. Throughout our experiments, the Day 1 dataset is exclusively used to train the neural network of baseline models. Both the baseline evaluation and the testing of the physics-informed prompts in LLMs are conducted on the unseen Day 2 dataset to rigorously assess generalization.

### 4.2 Baselines

We compare our method with representative baselines from three categories: **Deep learning models**, including MuSAC-C (Ji et al., 2026), LSTM (Siami-Namini et al., 2019), TCN (Gopali et al., 2021), and CsiNet (Wen et al., 2018); **Fine-tuning LLM**, GPT2-LLM (Cui et al., 2025); and **Mathematical methods**, including uniform sampling followed by linear interpolation (UL), polynomial regression, and 1D-DCT (Ahmed et al., 1974). We further evaluate several open-source LLMs with different parameter scales (Qwen3.6-27B, Gemma4-31B and Qwen3.5-397B-A17B) to assess the generality of our prompts across different LLMs.

### 4.3 Implementation settings

We establish two distinct settings (static scenario and dynamic scenario) alongside two compression ratios (1/16 and 1/32). To evaluate the upper-bound performance of our prompt-based framework, we evaluate our method on state-of-the-art flagship LLMs, including Gemini-3.1 Pro and GPT-5.4 Thinking. Unless otherwise specified, the WCR at time $t$ only references the previous one signal frame, i.e., $X_{t-1}$ at compressor and $\hat{X}_{t-1}$ at reconstructor. No historical information is provided for the first frame of each sequence. For each physical scenario, we selected four historical compression strategies that achieved a reconstruction NMSE superior to -25 dB. These top-performing schemes were then organized and embedded into the dictionary module of physics-informed prompt, as self-learned reference patterns. To comprehensively evaluate whether our method preserves task-relevant physical characteristics beyond basic reconstruction, we introduce two downstream sensing tasks: scenario classification (static vs. dynamic) and environment classification (indoor vs.

| | Method | Static Scenario | | Dynamic Scenario | | Avg. | |
|---|---|---|---|---|---|---|---|
| | | 1/16 | 1/32 | 1/16 | 1/32 | Method | Group |
| Deep Learning models | MuSAC-C | -4.34 | -4.27 | -5.02 | -5.22 | -4.71 | -5.63 |
| | LSTM | -4.91 | -5.10 | -4.78 | -5.03 | -4.96 | |
| | TCN | -5.35 | -5.57 | -5.19 | -6.60 | -5.68 | |
| | CsiNet | -7.02 | -6.38 | -7.58 | -7.77 | -7.19 | |
| Fine-tuning LLM | GPT2-LLM | -9.91 | -8.73 | -10.00 | -8.50 | -9.29 | -9.29 |
| Mathematical methods | UL | -10.88 | -6.39 | -13.43 | -11.52 | -10.56 | -11.30 |
| | Polynomial Regression | -11.50 | -7.44 | -13.48 | -13.45 | -11.47 | |
| | 1D-DCT | -11.74 | -7.75 | -14.35 | -13.64 | -11.87 | |
| Lightweight open-source LLMs | Qwen3.6-27B | -12.32 | -7.55 | -13.26 | -10.88 | -11.00 | -11.17 |
| | Gemma4-31B | -12.01 | -9.18 | -13.96 | -10.20 | -11.34 | |
| Large open-source LLMs | Qwen3.5-397B-A17B | -19.22 | -20.01 | -17.68 | -14.09 | -17.75 | -17.75 |
| Physics-informed Prompt | Gemini-3.1 Pro | **-20.67** | -21.59 | -20.22 | -20.33 | -20.70 | **-20.98** |
| | GPT-5.4 Thinking | -19.47 | **-23.10** | **-21.50** | **-20.97** | **-21.26** | |

Table 1: Performance comparison of the physics-informed prompt against baseline methods. The values presented in the table denote the average NMSE (dB) for each experimental setting. **Bold** denotes the best result.

| **Method** | **Task 1: Scenario Classification** | | | | | | | | | | **Task 2: Environment Classification** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Indoor 20MHz** | | **Outdoor 20MHz** | | **Indoor 40MHz** | | **Wi-BFI** | | **Task 1 AVG.** | | **Task 2 AVG.** | |
| | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE |
| Baseline (Original Data) | 100.00 | - | 99.39 | - | 94.38 | - | 100.00 | - | 98.23 | - | 100.00 | - |
| Deep Learning Models | 59.69 | -6.39 | 84.98 | -8.31 | 80.08 | -5.29 | 40.47 | -6.95 | 70.10 | -6.55 | 46.98 | -7.25 |
| Fine-tuning LLM | 68.12 | -9.68 | 98.48 | -10.39 | 91.25 | -6.87 | **93.12** | -9.16 | 87.06 | -8.76 | 83.61 | -9.80 |
| Mathematical Methods | 89.91 | -10.86 | 98.00 | -15.32 | 87.68 | -12.52 | 39.11 | -8.51 | 84.42 | -11.64 | 82.99 | -10.15 |
| **GPT5.4 Thinking** | **93.75** | **-21.27** | **100.00** | **-20.91** | **100.00** | **-21.10** | 87.50 | **-19.58** | **96.43** | **-20.71** | **100.00** | **-21.26** |

Table 2: Overall performance on two downstream sensing tasks. Specifically, Task 1 was evaluated across multiple datasets. Both Task accuracy (%) and NMSE (dB) are reported. Detailed performance for all individual methods are provided in Appendix B.2. **Bold** denotes the best result.

outdoor). The training and inference costs of the baselines and prompt-based methods are detailed in Appendix C.

## 5 Experimental Results

### 5.1 Overall Performances

The overall experimental results are displayed in Table 1. Our proposed physics-informed prompt outperforms all baseline methods by a significant margin. Compared with deep learning-based WCR methods, our approach achieves an average improvement of 15.35dB, effectively addressing the performance degradation commonly observed in conventional deep learning methods under distribution shifts. Distribution shifts are particularly prevalent in wireless signals: even within the same environment, different collection time can introduce noticeable drift, limiting the robustness of conventional deep-learning-based methods. In contrast, LLMs exhibits strong training-free generalization capability. Despite the modality gap, the proposed design effectively bridges it, enabling accurate reconstruction for newly collected signals without retraining. Furthermore, our method exhibits a 9.68dB improvement over the mathematical baselines. Unlike traditional mathematical methods that rely on fixed rules, the LLMs can reason over input data and prompt context to adaptively generate suitable data processing strategies tailored to each instance. In this sense, the LLM effectively acts as an **adaptive algorithm designer**, enabling flexible and data-aware signal processing beyond rigid handcrafted methods. Our methods shows reduced performance on lightweight models compared to flagship closed-source ones due to smaller LLMs' limited capacity for complex data processing and feature extraction. Nevertheless, the performance of Qwen3.5-397B-A17B remains close to flagship closed-source LLMs, indicating that our method achieves excellent performance on both open-source and closed-source models and verifying its practicality. In terms of prompt transferability, the physics-informed prompt achieves

| Prompt Configuration | Static Scenario | | Dynamic Scenario | |
|---|---|---|---|---|
| | 1/16 | 1/32 | 1/16 | 1/32 |
| **Gemini 3.1 Pro** | | | | |
| Baseline Prompt | -12.81 | -8.06 | -13.83 | -13.07 |
| w/ $\mathcal{K}$ | -14.06 | -11.11 | -16.16 | -13.66 |
| w/ $\mathcal{K}$ & $\mathcal{PI}$ & $X_{t-1}$ | -18.09 | -16.05 | -19.63 | -18.21 |
| w/ $\mathcal{K}$ & $\mathcal{PI}$ & $X_{t-1}$ & $\mathcal{D}$ | **-20.67** | **-21.59** | **-20.22** | **-20.33** |
| **GPT-5.4 Thinking** | | | | |
| Baseline Prompt | -12.97 | -7.54 | -13.79 | -11.50 |
| w/ $\mathcal{K}$ | -16.26 | -11.93 | -17.19 | -14.77 |
| w/ $\mathcal{K}$ & $\mathcal{PI}$ & $X_{t-1}$ | -18.61 | -15.29 | -19.64 | -16.40 |
| w/ $\mathcal{K}$ & $\mathcal{PI}$ & $X_{t-1}$ & $\mathcal{D}$ | **-19.47** | **-23.10** | **-21.50** | **-20.97** |

Table 3: Ablation study of different prompt components on WCR performance. 'w/' is an abbreviation for 'with', indicating the incremental addition of components to the baseline prompt. The '&' symbol denotes the combination of multiple integrated modules. $\mathcal{K}$, $\mathcal{PI}$, and $\mathcal{D}$ represent key prompt module, physics-informed priors module, and dictionary module, respectively.

exceptional results on both flagship LLMs, with only a 0.56dB performance gap between Gemini-3.1 Pro and GPT-5.4 Thinking. This minimal variance indicates that our physics-informed prompt design is highly model-agnostic. The results in Table 2 show that our method outperforms the baseline approaches in both sensing tasks and reconstruction performance across most datasets. These results demonstrate that our method not only reconstructs the data with high fidelity, but also preserves the underlying physical characteristics required by downstream sensing tasks. This further indicates our methods' strong generalization ability across different experimental scenarios and datasets.

From the perspective of scenario robustness, the performance discrepancy between the static and dynamic scenarios is marginal (only 0.45dB), proving that our designed prompts are highly robust and adaptable to diverse physical settings. From the perspective of compression ratio, the results of physics-informed prompt are sometimes even better under the higher compression ratio. This observation suggests that LLMs possess strong reasoning capabilities: under more extreme compression conditions, they are encouraged to infer more effective reconstruction strategies, rather than simply depending on preserving more original values in the compressed matrices.

## 5.2 Ablation Study

In this section, we analyze the contribution of each component in our proposed prompting framework. We begin with a baseline prompt that only contains the basic description of the dataset and task (shown as the uncolored text in Figure 1), without any of the designed components. These components are then incrementally added to quantify their individual contributions. Furthermore, we conduct a fine-grained ablation study on the sub-modules of the physics-informed priors, including scenario information and scenario-specific knowledge, together with the previous wireless signal frames. These studies aim to reveal how physical information actively helps the LLMs comprehend the wireless signals to facilitate reconstruction.

| Prompt Configuration | Static Scenario | | Dynamic Scenario | |
|---|---|---|---|---|
| | 1/16 | 1/32 | 1/16 | 1/32 |
| w/ $\mathcal{K}$ | -14.06 | -11.11 | -16.16 | -13.66 |
| w/ $\mathcal{K}$ & $\mathcal{S}$ | -15.62 | -13.32 | -18.21 | -14.89 |
| w/ $\mathcal{K}$ & $\mathcal{S}$ & $\mathcal{P}$ | -16.52 | -14.28 | -18.70 | -15.62 |
| w/ $\mathcal{K}$ & $\mathcal{S}$ & $\mathcal{P}$ & $X_{t-1}$ | **-18.09** | **-16.05** | **-19.64** | **-18.21** |

Table 4: Fine-grained ablation study on the sub-modules of the physics-informed priors module and previous wireless signal frame. $\mathcal{S}$ and $\mathcal{P}$ represent the scenario information sub-module and scenario-specific knowledge sub-module, respectively.

**Effects of Key Prompt** Independently deployed compressor and reconstructor LLMs often suffer from semantic misalignment, hindering the reconstructor from extracting valuable recovery cues from compressed representations. To bridge this gap, we instruct the compressor to embed essential scenario analysis and recovery guidelines into a shared key prompt (e.g., *"Crucial reconstruction hints must be included in the key prompt"*). This explicit guidance enables the reconstructor to accurately interpret the compressed data. As shown in Table 3, this module yields an average performance improvement of 2.69dB, demonstrating its effectiveness in mitigating cross-model semantic misalignment.

**Effects of Physics-Informed Priors** We inject physics-informed priors ($\mathcal{PI}$) to guide LLMs toward high-fidelity, physically consistent reconstruction. As shown in Table 3, combining $\mathcal{PI}$ with the previous frame ($X_{t-1}$) yields a 3.34dB improvement, demonstrating that structured priors empower LLMs to uncover deep hidden signal features for WCR.

Table 4 further ablates the $\mathcal{PI}$ sub-modules. Introducing scenario information ($\mathcal{S}$) provides a 1.76dB gain, indicating that even in a training-free setting, LLMs can effectively associate raw numerical sequences with their physical environ-

| Transmission Method | Static Scenario | | Dynamic Scenario | | AVG. |
|---|---|---|---|---|---|
| | 1/16 | 1/32 | 1/16 | 1/32 | |
| Full key-prompt | **-19.47** | -23.1 | **-21.5** | -20.97 | -21.26 |
| Codebook-index | -19.03 | **-25.58** | -19.69 | **-22.49** | **-21.69** |

Table 5: Performance comparison between full key-prompt transmission and codebook-index transmission.

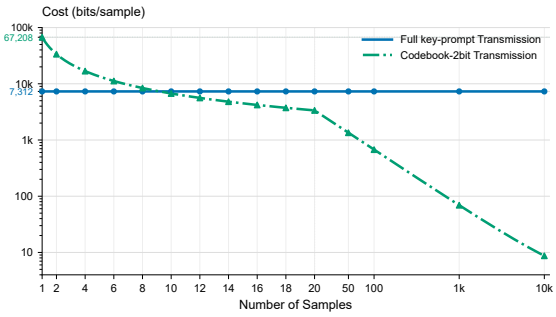



Figure 3: Amortized transmission cost of the key prompt via indexed codebook representation.

ments. However, establishing such associations relies on the LLMs' ability to infer the relationship from its internal knowledge. To mitigate this, we supply explicit scenario-specific knowledge ($\mathcal{P}$) that directly links environmental traits to temporal and frequency-domain signal variations. This fine-grained expert guidance yields an additional 0.77dB gain, confirming the superiority of explicit domain instruction over implicit knowledge retrieval.

**Effects of Previous Wireless Signal Frame** Without historical context, LLMs rely exclusively on frequency-domain correlations for compression. However, providing the previous frame ($X_{t-1}$) enables them to exploit rich temporal dependencies. As shown in Table 4, integrating $X_{t-1}$ yields an additional 1.72dB improvement. This demonstrates that by leveraging scenario-specific knowledge and their inherent time-series processing capabilities, LLMs can effectively fuse historical context with physical priors to formulate more efficient, temporally-aware WCR strategies. We further evaluate the impact of reference frame length in Appendix B.3, where the results show that using only a using a single previous frame achieves the best performance, while excessive historical frames may introduce historical bias and degrade WCR performance.

**Effects of Dictionary Module** LLMs exhibit significant inherent stochasticity during long-context CoT generation, where cumulative error propagation can severely degrade deterministic tasks like WCR. To mitigate this and ensure practical output stability, we introduce a dictionary module comprising a subset of high-performing, previously generated compression strategies. As shown in Table 3, integrating this module improves the average reconstruction performance by 2.71 dB. This demonstrates that the dictionary module effectively balances unconstrained exploration with rigid exploitation, anchoring the LLMs toward optimal and stable WCR pathways.

We further provide a detailed analysis in Appendix B.4. The results show that the dictionary module is particularly effective in static scenarios, where wireless signals exhibit more regular physical patterns and can therefore benefit more from explicit strategy guidance. In highly dynamic scenarios, however, the dictionary module may introduce a mild strategy bias by encouraging LLMs to over-rely on predefined compression strategies, reflecting a trade-off between output stability and adaptive exploration.

### 5.3 Codebook Representation of Key Prompt

We designate the semantic centroid of each key prompt cluster as the representative exemplar for the corresponding experimental environment. These exemplars are used to construct a codebook, which is then deployed at both the compressor and the reconstructor. Consequently, during inference, the LLM only needs to output an index representing the selected exemplar. We evaluate the effectiveness of our framework when transmitting only a 2-bit index (i.e., maintaining up to four exemplars per experimental scenario).

Table 5 demonstrates that codebook-based representation achieves overall performance comparable to transmitting the full key prompt, and even yields marginal improvements in certain cases. This can be attributed to the method's explicit constraints on the LLM's compression strategy, which lead to more stable and robust performance in scenarios with high compression ratios. Conversely, in low-compression-ratio settings, relying on a fixed key prompt limits the ability to incorporate fine-grained, instance-specific compression details, resulting in a slight performance degradation. Furthermore, Figure 3 illustrates that when amortized over a large volume of samples, the per-sample communication overhead of the key prompt under codebook-based

representation is substantially lower than that of the full transmission approach. This underscores the effectiveness of the method in mitigating the communication cost associated with key prompts.

## 6 Conclusion

We introduce a training-free, physics-informed prompting framework for WCR that overcomes the generalization and physical fidelity limitations of conventional methods. By integrating structured physical priors with two coordination mechanisms—a dictionary module for stabilization and a key-prompt module for compressor-reconstructor alignment—our framework enables LLMs to robustly process rigid, physics-constrained signal data. Experiments confirm that our approach maintains physical consistency, mitigates generation stochasticity, and resolves semantic misalignments. Ultimately, this work demonstrates the compelling potential of extending LLM prompting beyond natural language to physics-constrained, low-level signal processing tasks.

## Limitations

**Large inference cost of LLMs** Current prompt-based methods require substantial inference time and token cost for WCR, which may limit their practicality in real-world scenarios. In future work, we plan to further optimize the prompt design to reduce inference latency and token overhead while maintaining reconstruction performance.

**Constrained creativity of LLMs** The dictionary module may bound the LLM's generative search space in certain highly specialized scenarios so that LLMs rely excessively on predefined patterns in the dictionary module instead of actively reasoning about more effective compression or reconstruction strategies. Future work should investigate more adaptive prompt designs that can better balance exploration and exploitation for LLMs in more complex scenarios.

**More modalities and fine-grained priors** We focus on Wi-Fi signals for both method design and empirical evaluation. Wi-Fi signals share common properties with other wireless signal modalities, such as CSI and RF signals, including compressibility, rich physical characteristics, and strong sensing capabilities. Therefore, we believe that the core design of our method can be broadly extended to other types of wireless signals. We leave a more comprehensive evaluation on diverse wireless signal modalities as future work. Meanwhile, we currently categorize environments into coarse-grained static and dynamic scenarios. In future work, we will introduce more fine-grained environment-related physical priors to further evaluate our method.

## Ethical considerations

Data collection for our self-collected Wi-CBR dataset only involved the authors of this manuscript. No external human participants were recruited. All authors have agreed the utilization of dataset. The dataset doesn't contain any human image and other personally identifiable information. All experiments involving human subjects have been approved by our IRB.

## References


Nasir Ahmed, T. Natarajan, and K. R. Rao. 1974. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93.

Tuo An, Yunjiao Zhou, Han Zou, and Jianfei Yang. 2025. IoT-LLM: A framework for enhancing large language model reasoning from real-world sensor data. *Preprint*, arXiv:2410.02429.

Omer Aydin, Enis Karaarslan, Fatih Safa Erenay, and Nebojsa Bacanin. 2025. Generative AI in academic writing: A comparison of deepseek, qwen, chatgpt, gemini, llama, mistral, and gemma. *Preprint*, arXiv:2503.04765.

Yiming Cui, Jiajia Guo, Chao-Kai Wen, Shi Jin, and En Tong. 2025. Leveraging pre-trained large language models for CSI feedback in massive MIMO systems. *Authorea Preprints*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Saroj Gopali, Faranak Abri, Sima Siami-Namini, and Akbar Siami Namin. 2021. A comparison of TCN and LSTM models in detecting anomalies in time series data. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2415–2420.

Jiajia Guo, Xiangyi Li, Muhan Chen, Peiwen Jiang, Tingting Yang, Weiming Duan, Haowen Wang, Shi Jin, and Quan Yu. 2020. Ai enabled wireless communications with real channel measurements: Channel feedback. *Journal of Communications and Information Networks*, 5(3):310–317.

Khandaker Foysal Haque, Francesca Meneghello, and Francesco Restuccia. 2023. Wi-BFI: Extracting the IEEE 802.11 beamforming feedback information from commercial Wi-Fi devices. In *Proceedings of the 17th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '23, page 104–111, New York, NY, USA. Association for Computing Machinery.

Jingzhi Hu, Tianyue Zheng, Zhe Chen, Hongbo Wang, and Jun Luo. 2023. MUSE-Fi: Contactless mutiperson sensing exploiting near-field Wi-Fi channel variation. In *Proceedings of the 29th annual international conference on mobile computing and networking*, pages 1–15.

Sijie Ji, Lixiang Lian, Yuanqing Zheng, and Chenshu Wu. 2024. MuSAC: Mutualistic sensing and communication for mobile crowdsensing. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pages 243–254.

Sijie Ji, Yiwei Wang, and Lixiang Lian. 2026. MuSAC: Toward communication-free sensory data acquisition. *IEEE Transactions on Mobile Computing*, pages 1–18.

Xin Li, Jingzhi Hu, and Jun Luo. 2024. Efficient beamforming feedback information-based Wi-Fi sensing by feature selection. *IEEE Wireless Communications Letters*, 13(9):2347–2351.

Boxun Liu, Xuanyu Liu, Shijian Gao, Xiang Cheng, and Liuqing Yang. 2024. LLM4CP: Adapting large language models for channel prediction. *Journal of Communications and Information Networks*, 9(2):113–125.

Yili Ren, Haopeng Zhang, Haohan Yuan, Jingzhe Zhang, and Yitong Shen. 2025. Wi-Chat: Large language model-powered wi-fi-based human activity recognition. EnvSys '25, page 8–13, New York, NY, USA. Association for Computing Machinery.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Haotian Tian and Lixiang Lian. 2024. Gsure-based unsupervised deep equilibrium model learning for large-scale channel estimation. In *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, pages 4938–4943.

Hongbo Wang, Jingyang Hu, Tianyue Zheng, Jingzhi Hu, Zhe Chen, Hongbo Jiang, Yuanjin Zheng, and Jun Luo. 2024. MuKI-Fi: Multi-person keystroke inference with BFI-Enabled Wi-Fi sensing. *IEEE Transactions on Mobile Computing*, 23(10):9835–9850.

Chao-Kai Wen, Wan-Ting Shih, and Shi Jin. 2018. Deep learning for massive MIMO CSI feedback. *IEEE Wireless Communications Letters*, 7(5):748–751.

Chenhao Wu, Xuan Huang, Jun Huang, and Guoliang Xing. 2023. Enabling ubiquitous WiFi sensing with beamforming reports. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, page 20–32, New York, NY, USA. Association for Computing Machinery.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Huanqi Yang, Rucheng Wu, and Weitao Xu. 2024. Transcompressor: LLM-powered multimodal data compression for smart transportation. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 2335–2340.

Man Yuan, Lixiang Lian, and Sijie Ji. 2025. Physics-Informed data augmentation for CSI data-limited deep learning-driven communication tasks. In *2025 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1634–1639. IEEE.

Yunming Zhao, Wei Gong, Minghui Liwang, Li Li, Baoxian Zhang, and Cheng Li. 2025. A generalizable prompt-based prototypical framework for CSI-based few-shot and cross-domain activity recognition. *IEEE Transactions on Mobile Computing*, 24(12):13125–13141.

# A Experimental Settings

## A.1 Dataset

**Wi-CBR Data Collection Process** The Wi-CBR dataset comprises wireless signals collected in a real-world environment. Specifically, a Wi-Fi communication link was established between a mobile smartphone and a router. To capture the wireless signal transmissions between the devices, we deployed a wireless sniffer consisting of a Wi-Fi 6 network interface card and an NVIDIA Jetson AGX Xavier mobile computing node, as shown in Figure 4a. The indoor experimental scenario is depicted in Figure 4b. To introduce wireless signal interference in dynamic scenario, a person continuously walked back and forth across the main communication path between the smartphone and the router, as shown in Figure 4c. The transmission rate of the wireless signals is approximately 8 to 10 frames per second. Consequently, the extremely short time interval between adjacent frames results in strong temporal continuity.

**Dataset Categories** To further evaluate the generalization ability of our framework across different experimental scenarios and datasets, we categorized the self-collected datasets into four types based on different experimental parameter settings. Additionally, we introduced a public dataset, Wi-BFI (Haque et al., 2023). The specific experimental parameters for these datasets are detailed as follows:

- **SC-CBR (Indoor, 20MHz):** This dataset was self-collected in an indoor scenario with a channel bandwidth of 20MHz. The data was gathered using continuous sampling. Unless otherwise specified, this serves as the default dataset for our evaluation experiments.

- **SC-CBR (Outdoor, 20MHz):** This dataset was self-collected in an outdoor scenario with a channel bandwidth of 20MHz. The data was gathered using continuous sampling.

- **SC-CBR (Indoor, 40MHz):** This dataset was self-collected in an indoor scenario with a channel bandwidth of 40MHz. The data was gathered using continuous sampling.

- **SC-CBR (Frame-skipping sampling, Indoor, 20MHz):** This dataset was self-collected in an indoor scenario with a channel bandwidth of 20MHz. The data was gathered using frame-skipping sampling.

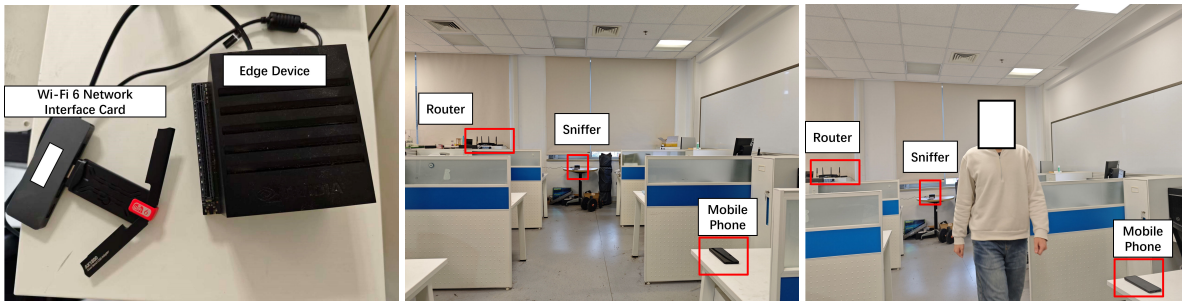
- **Wi-BFI (Indoor, 20MHz):** This is a public dataset collected in an indoor Wi-Fi environment with a channel bandwidth of 20MHz, using continuous sampling. Since this dataset does not explicitly distinguish between static and dynamic scenarios, we categorized it as a static scenario based on the magnitude heatmap of the wireless signals. Therefore, Wi-BFI does not contain dynamic scenario data.

**Details of Wireless Signals** In our experiment, wireless signals specifically refer to the feedback matrices within the Action No ACK data frames of the IEEE 802.11ax protocol. Because these feedback matrices are mathematically compressed versions of wireless communication beamforming reports, they are commonly referred to as **C**ompressed **B**eamforming **R**eports (CBR). CBR include rich physical information regarding the environment surrounding the wireless communication link. Our Wi-CBR dataset consists of sequential CBRs extracted from these Action No ACK frames, where each sequence contains five temporally consecutive frames. Under 20MHz communication bandwidth, each CBR has dimensions of $64 \times 10$, where the 64 rows correspond to 64 distinct frequency subcarriers, indicating strong spatial features within each frame. Under 40 MHz bandwidth setting, each CBR has dimension of $122 \times 10$. The physical information for each subcarrier is represented by 10 integer angle values ranging from 0 to 16 or 0 to 64.

## A.2 Implementation Details

- MuSAC-C (Ji et al., 2026): MuSAC-C is a multilayer neural architecture specifically designed for WCR. However, it processes wireless signals independently and does not account for the temporal correlations between adjacent frames. We use the same network architecture and experimental settings as described in the original paper.

- LSTM (Siami-Namini et al., 2019): LSTM is widely used network architectures for modeling temporal dependencies in sequential data. We use a two-layer unidirectional LSTM network, while the remaining components




(a) Hardware configuration of sniffer (b) Static scenario (c) Dynamic scenario

Figure 4: Illustration of the Wi-CBR dataset collection scenarios.

are implemented with multi-layer perceptrons (MLPs) to align the data dimensions. The learning rate is set to $1 \times 10^{-3}$, the L2 regularization coefficient is set to $1 \times 10^{-5}$, the number of training epochs is 10,000, and the batch size is 32.

- TCN (Gopali et al., 2021): TCN is a sequence modeling architecture based on causal and dilated 1D convolutions, which can capture long-range temporal dependencies while preserving temporal order. Existing studies (Zhao et al., 2025) have applied them for wireless sensing tasks. We introduce a causal convolution module to ensure that the output at the current frame depends only on the current and historical frames. The remaining components are also implemented with MLPs for dimensional alignment. respectively. The experimental hyperparameters and settings for TCN are the same as those used for LSTM.

- CsiNet (Wen et al., 2018): CsiNet is an autoencoder-based neural network benchmark for CSI compression and reconstruction, where CSI is a common wireless signal representation used in wireless sensing applications. We adapt it to CBR data by using convolutional layers to extract local features from each two-dimensional CBR frame, followed by a fully connected layer that compresses the features into a low-dimensional codeword. The decoder projects the codeword back to the feature space and uses RefineNet-style residual convolutional blocks to reconstruct the CBR frame.

- GPT2-LLM (Cui et al., 2025): GPT2-LLM is a fine-tuning-based LLM baseline that fine-tunes the intermediate layers of a pre-trained GPT-2 model for WCR. The encoder applies a fixed Gaussian random projection to compress each CBR frame into a low-dimensional codeword, while the decoder first obtains a coarse reconstruction through pseudo-inverse projection and then maps both the coarse reconstruction and the compressed codeword into continuous pseudo-token embeddings for GPT-2. During training, most GPT-2 parameters are frozen, and only the layer normalization parameters, positional embeddings, input bridge, and output reconstruction head are updated.

- Uniform sampling followed by linear interpolation (UL): UL uniformly down-samples the rows of each wireless signal frame during the compression phase based on compression ratio, subsequently recovering the missing data via linear interpolation during reconstruction.

- Polynomial regression: Polynomial regression applies a column-wise regression to each wireless signal frame: it employs a cubic polynomial fit (retaining 4 data points per column) at a 1/16 compression ratio, and a first-order linear fit (retaining 2 data points per column) at a 1/32 compression ratio.

- 1D-DCT (Ahmed et al., 1974): 1D-DCT is a classical mathematical method widely used for digital signal compression. It decomposes each CBR column's 1D signal into cosine components of varying frequencies, compressing it by retaining only low-frequency components. Reconstruction is performed by zero-padding the discarded high-frequency coefficients and applying inverse DCT, requiring no model training.

We collected 2,540 wireless data frames to train

| Dataset | Static Scenario | | Dynamic Scenario | | AVG. |
|---|---|---|---|---|---|
| | 1/16 | 1/32 | 1/16 | 1/32 | |
| SC-CBR (Indoor, 20MHz) | -19.47 | -23.1 | -21.5 | -20.97 | -21.27 |
| SC-CBR (Outdoor, 20MHz) | -21.70 | -18.48 | -25.41 | -18.01 | -20.91 |
| SC-CBR (Indoor, 40MHz) | -25.22 | -17.87 | -24.59 | -16.71 | -21.09 |
| SC-CBR (Frame-skipping sampling,Indoor, 20MHz) | -21.15 | -22.27 | -21.50 | -20.63 | -21.38 |
| Wi-BFI (Indoor, 20MHz) | -19.19 | -19.97 | – | – | -19.58 |

Table 6: Performance comparison across different datasets.

the deep learning baseline models, partitioning them into training and validation sets at a ratio of 4:1. For the evaluation of our proposed physics-informed prompt, we use the default settings on the official website[2345]. To ensure the independence of each evaluation, we disable the historical memory feature and initiate a new conversation session for every compression or reconstruction experiment. Our baseline experiments are conducted on a server equipped with an Intel Xeon Gold 6354 CPU and 512 GB RAM. The software environment includes CUDA 12.6 and PyTorch 2.6.0. For the first downstream task, scenario classification (static vs. dynamic), we utilized four datasets (Indoor 20MHz, Outdoor 20MHz, Indoor 40MHz, and Wi-BFI), assigning the CBR data from static scenarios across all datasets as positive labels and the data from dynamic scenarios as negative labels. For the second downstream task, environment classification (indoor vs. outdoor), we categorized the Indoor 20MHz and Wi-BFI datasets as indoor environment samples, and the Outdoor 20MHz dataset as outdoor environment samples. Our downstream task network employs a lightweight CNN architecture, where the model takes two consecutive frames from the same data sequence as input and outputs a binary classification label. The downstream task network is trained on all labeled datasets and used as a benchmark to evaluate the sensing-task performance of the CBR reconstructed by different methods. To ensure robust evaluation, we implement a post-processing procedure for invalid LLM outputs, such as mismatched output dimensions. Specifically, we process such outputs by truncating them to the valid dimension or padding them using the last row. For more complex output-format errors, we re-query the API to obtain new results.

[2]Gemini-3.1 Pro: https://gemini.google.com/app

[3]GPT-5.4 Thinking: https://chatgpt.com/

[4]Qwen: https://chat.qwen.ai/

[5]Gemma4-31B: https://aistudio.google.com/prompts/new_chat

# B Evaluation

## B.1 Generalization Across Different Datasets

Overall, the results in Table 6 demonstrate that our framework has strong generalizability across different environments, bandwidth, sampling strategy and other public datasets. However, we observe a slight performance degradation at the 1/32 compression ratio in outdoor scenarios. This can be attributed to the increased complexity and volatility of outdoor physical environments, which challenge the LLMs' ability to compress and reconstruct wireless signals under extreme compression constraints. A similar result is also evident in the 40 MHz setting, where the expanded volume of subcarrier information makes it difficult for LLMs to extract sufficient essential features to facilitate high-fidelity reconstruction at such high compression ratios. Although our current evaluation encompasses various fine-grained signal features—including distinct data collection scenarios, sampling strategies, and communication bandwidths—the real-world wireless signal space remains vastly diverse. Future work will explore incorporating a broader spectrum of signal characteristics, such as diverse modulation schemes and specific hardware effects, to further validate the generalizability of LLM-based signal reconstruction.

## B.2 Performance on Downstream Tasks

As shown in Table 7, we report both the downstream task performance and the reconstruction quality of all baselines on Task 1. Our method consistently outperforms the baseline approaches in both sensing accuracy and reconstruction performance across most datasets. These results demonstrate that our method not only reconstructs the data with high fidelity, but also preserves the underlying physical characteristics required by downstream

| Method | Pairs | Indoor 20MHz | | | | Outdoor 20MHz | | | | Indoor 40MHz | | | | Wi-BFI | | | | Method AVG. | | Group AVG. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1/16 | | 1/32 | | 1/16 | | 1/32 | | 1/16 | | 1/32 | | 1/16 | | 1/32 | | Acc | NMSE | Acc | NMSE |
| | | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | Acc | NMSE | | | | |
| Original Data | 564 | 100.00 | - | 100.00 | - | 99.39 | - | 99.39 | - | 94.38 | - | 94.38 | - | 100.00 | - | 100.00 | - | 98.23 | - | 98.23 | - |
| MuSAC-C | 564 | 63.12 | -6.95 | 68.12 | -6.73 | 92.68 | -9.10 | 68.29 | -8.66 | 60.00 | -5.31 | 60.00 | -5.04 | 13.75 | -7.02 | 7.50 | -6.90 | 60.55 | -6.76 | 59.20 | -6.55 |
| LSTM | 564 | 50.00 | -5.32 | 50.00 | -5.34 | 48.78 | -7.07 | 48.78 | -7.06 | 50.00 | -4.62 | 50.00 | -4.64 | **100.00** | -5.46 | **100.00** | -5.42 | 56.74 | -5.55 | | |
| TCN | 564 | 68.12 | -5.66 | 60.62 | -5.92 | 58.54 | -8.21 | 70.73 | -8.44 | 70.00 | -4.85 | 68.12 | -4.72 | 26.25 | -7.29 | 38.75 | -7.79 | 61.26 | -6.26 | | |
| CsiNet | 564 | 62.50 | -7.84 | 55.00 | -7.36 | 89.63 | -9.36 | 76.83 | -8.59 | 50.00 | -6.62 | 53.75 | -6.48 | 21.25 | -7.86 | 16.25 | -7.83 | 58.24 | -7.64 | | |
| GPT2-LLM | 564 | 71.25 | -10.14 | 65.00 | -9.21 | 95.73 | -10.00 | 95.73 | -10.77 | 59.38 | -6.97 | 69.38 | -6.77 | 90.00 | -9.19 | 96.25 | -9.12 | 78.63 | -8.76 | 78.63 | -8.76 |
| UL | 564 | 95.62 | -12.39 | 63.75 | -8.67 | **100.00** | -16.35 | 86.59 | -12.00 | 93.75 | -14.65 | 81.25 | -11.06 | 92.50 | -10.53 | 33.75 | -5.43 | 83.51 | -11.32 | | |
| Polynomial Regression | 564 | **96.88** | -12.01 | 90.00 | -9.60 | **100.00** | -16.98 | **100.00** | -14.59 | 88.75 | -12.10 | 77.50 | -9.35 | 46.25 | -9.94 | 1.25 | -7.78 | 82.54 | -11.32 | 84.01 | -11.64 |
| 1D-DCT | 564 | 96.25 | -12.62 | 91.88 | -9.84 | 99.39 | -17.40 | **100.00** | -14.62 | 94.38 | -15.19 | 90.62 | -12.74 | 55.00 | -9.58 | 2.50 | -7.78 | 85.99 | -12.27 | | |
| **GPT-5.4 Thinking** | 56 | 87.50 | **-20.49** | **100.00** | **-22.04** | **100.00** | **-23.56** | **100.00** | **-18.25** | **100.00** | **-24.91** | **100.00** | **-17.29** | 75.00 | **-19.19** | **100.00** | **-19.97** | **96.43** | **-20.71** | **96.43** | **-20.71** |

Table 7: Comprehensive performance summary for Task 1: scenario classification. Pairs denotes the number of test samples evaluated for each method. Task accuracy (%) and NMSE (dB) are reported across four datasets under 1/16 and 1/32 compression ratios. **Bold** denotes the best result.

| Group | Method | Pairs | Indoor 20MHz | | | | Method AVG. | | Group AVG. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1/16 | | 1/32 | | Acc | NMSE | Acc | NMSE |
| | | | Acc | NMSE | Acc | NMSE | | | | |
| Baseline | Original Data | 244 | 100.00 | - | 100.00 | - | 100.00 | - | 100.00 | - |
| Lightweight open-source LLMs | Qwen3.6-27B | 8 | 75.00 | -12.79 | 50.00 | -9.22 | 62.50 | -11.00 | 68.75 | -11.17 |
| | Gemma-31B | 8 | 100.00 | -12.99 | 50.00 | -9.69 | 75.00 | -11.34 | | |
| Large open-source LLMs | Qwen3.5-397B-A17B | 8 | 100.00 | -18.45 | 100.00 | -17.05 | 100.00 | -17.75 | 100.00 | -17.75 |
| Physics-informed Prompt | Gemini-3.1 Pro | 8 | 100.00 | -20.45 | 100.00 | -20.96 | 100.00 | -20.70 | **100.00** | **-20.98** |
| | GPT-5.4 Thinking | 16 | 100.00 | -20.49 | 100.00 | -22.04 | 100.00 | -21.26 | | |

Table 8: Performance summary of LLM-based methods for the Indoor 20MHz dataset in Task1: scenario classification. Pairs denotes the number of test samples evaluated for each method. Task accuracy (%) and NMSE (dB) are reported under 1/16 and 1/32 compression ratios.

sensing tasks. This further indicates its strong generalization ability across different experimental scenarios and datasets.

We also observe that the baseline methods exhibit substantial performance fluctuations across datasets, especially on the Wi-BFI dataset. This is because the training and testing data in this setting contain only static samples. When the reconstruction quality is poor, the temporal correlation between adjacent CBR frames can be disrupted, causing the downstream classifier to incorrectly identify static samples as dynamic ones and leading to a significant performance drop. The LSTM baseline achieves 100% sensing accuracy on the Wi-BFI dataset because its reconstructed adjacent CBR frames tend to become nearly identical; as a result, the downstream classifier assigns all frame pairs to the static category, producing an abnormal performance increase. In contrast, the performance of LSTM on other datasets is close to 50%, which further supports the above observation. These results suggest that the downstream sensing performance of existing baseline methods is highly dependent on the dataset composition, scene dynamics, and architectural characteristics of the models themselves, making them less generalizable across diverse scenarios.

Since all LLM-related experiments in our study were conducted through web-based interactions, the amount of test data that could be processed by LLMs was limited. Therefore, we selected GPT-5.4 Thinking as the representative model of our method and evaluated it on all datasets. The results show that it still significantly outperforms all baseline methods, demonstrating the effectiveness and strong generalization capability of our approach. For other large language models, we conducted evaluations on the Indoor 20MHz dataset, as shown in Table 8. The results indicate that our method achieves strong downstream sensing performance when implemented with different flagship large-scale language models.

Table 9 further presents the downstream task performance and reconstruction quality of all baselines on Task 2. The results show that our method achieves the best performance across different sensing tasks. Since the reconstructed CBR data produced by each LLM in the indoor scenario were reused for both sensing tasks, the corresponding LLM-based results are identical across the two

| Group | Method | Pairs | Environment Classification | | | | Method AVG. | | Group AVG. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1/16 | | 1/32 | | Acc | NMSE | Acc | NMSE |
| | | | Acc | NMSE | Acc | NMSE | | | | |
| Baseline | Original Data | 244 | 100.00 | - | 100.00 | - | 100.00 | - | 100.00 | - |
| Deep Learning models | MuSAC-C | 244 | 50.00 | -7.58 | 43.44 | -7.40 | 46.72 | -7.49 | 46.98 | -7.25 |
| | LSTM | 244 | 34.43 | -6.37 | 34.43 | -6.35 | 34.43 | -6.36 | | |
| | TCN | 244 | 59.43 | -7.08 | 55.33 | -7.37 | 57.38 | -7.22 | | |
| | CsiNet | 244 | 52.87 | -8.11 | 45.90 | -7.76 | 49.39 | -7.94 | | |
| Fine-tuning LLM | GPT2-LLM | 244 | 88.93 | -9.76 | 78.28 | -9.83 | 83.61 | -9.80 | 83.61 | -9.80 |
| Mathematical methods | UL | 244 | 98.36 | -11.91 | 71.72 | -7.47 | 85.04 | -9.69 | 82.99 | -10.15 |
| | Polynomial Regression | 244 | 90.98 | -11.46 | 70.08 | -9.19 | 80.53 | -10.32 | | |
| | 1D-DCT | 244 | 95.08 | -11.57 | 71.72 | -9.29 | 83.40 | -10.43 | | |
| Lightweight open-source LLMs | Qwen3.6-27B | 8 | 75.00 | -12.79 | 50.00 | -9.22 | 62.50 | -11.00 | 68.75 | -11.17 |
| | Gemma-31B | 8 | 100.00 | -12.99 | 50.00 | -9.69 | 75.00 | -11.34 | | |
| Large open-source LLMs | Qwen3.5-397B-A17B | 8 | 100.00 | -18.45 | 100.00 | -17.05 | 100.00 | -17.75 | 100.00 | -17.75 |
| Physics-informed Prompt | Gemini-3.1 Pro | 8 | 100.00 | -20.45 | 100.00 | -20.96 | 100.00 | -20.70 | **100.00** | **-20.98** |
| | GPT-5.4 Thinking | 16 | 100.00 | -20.49 | 100.00 | -22.04 | 100.00 | -21.26 | | |

Table 9: Comprehensive performance summary for Task 2: Environment classification. Pairs denotes the number of test samples evaluated for each method. Task accuracy (%) and NMSE (dB) are reported under 1/16 and 1/32 compression ratios.

Table 10: Impact of reference wireless signal frame length ($K$) on WCR performance.

| Reference Length | Static Scenario | | Dynamic Scenario | |
|---|---|---|---|---|
| | 1/16 | 1/32 | 1/16 | 1/32 |
| $K = 0$ | -16.52 | -14.95 | -18.70 | -15.62 |
| $K = 1$ | **-20.67** | **-21.59** | **-20.22** | **-20.33** |
| $K = 2$ | -18.20 | -17.89 | -18.39 | -16.02 |

tasks. In future work, we plan to adopt more flexible API-based evaluation settings to further expand the scale of our experimental datasets.

### B.3 Impacts of Reference Frame Length

We evaluate the impact of varying reference frame lengths on WCR performance. As shown in Table 10, referencing only a single historical frame ($K = 1$) achieves the optimal performance. The performance gain over no reference frame is 4.26dB on average. Increasing the reference frame length does not monotonically improve performance. Our empirical analysis reveals that when supplied with an extended context of reference frames, LLMs exhibit a strong historical bias. They tend to over-rely on previous wireless signal frames to formulate temporal-based compression strategies, neglecting the unique, crucial data variations intrinsic to the current wireless signal frame itself. This behavior becomes more pronounced in dynamic scenarios, where the temporal correlations between adjacent frames are inherently weaker and less predictable. Consequently, incorporating excessive historical reference frames introduces unstable variables that biases the compression policy, leading to performance degradation. Moreover, as the temporal-based recovery mechanism is intrinsically susceptible to cumulative error propagation, such historical misguidance causes the WCR performances to deteriorate progressively over time.

### B.4 Further Analysis of the Dictionary Module

We further analyze the impact of the dictionary module on the stability and adaptability of WCR. From Figure 5, we observe that integrating the dictionary module under both compression ratios in the static scenario yields a significantly tighter performance distribution. Furthermore, performance gains in static scenario are substantially more obvious than those observed in the dynamic scenario. This discrepancy suggests that the underlying physical patterns of wireless signals in static environments are relatively straightforward. Consequently, when supplied with explicit methodological guidance via the dictionary module, the LLM can effortlessly capture the critical feature required to accomplish high-fidelity reconstruction.

In the dynamic scenario at a 1/16 compression ratio, we observe a different result. Although the integration of dictionary module improves both the average performance and overall stability, the

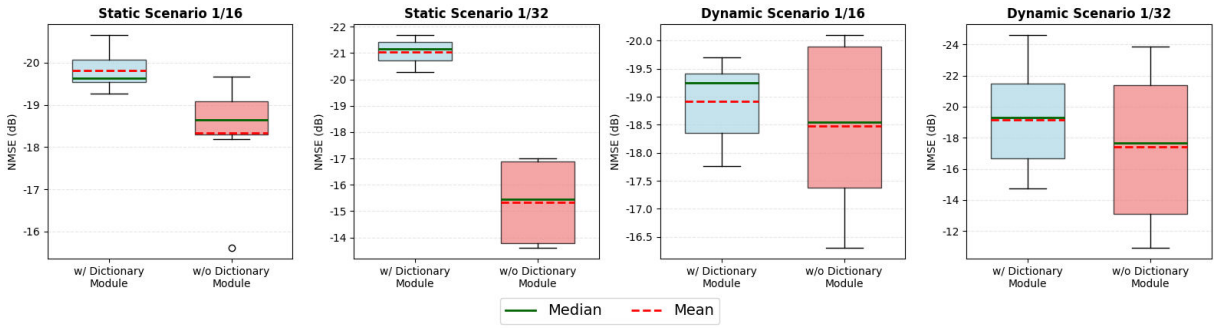



Figure 5: Impact of the dictionary module on reconstruction performance under different scenarios and compression ratios. Box plots show the distribution of NMSE (dB) with and without the dictionary module.

box plots reveal that the peak reconstruction quality (i.e., the upper bound and upper quartile) of the prompt without the dictionary module actually performances better. This interesting observation reveals two key phenomena. First, due to the highly dynamic variations of wireless signals in both temporal and frequency domains, the resulting signal patterns can be extremely complex. In such scenarios, a few set of stabilizing anchors is insufficient to encompass the complex variations of wireless signals. Second, it is important to note that the dictionary module only serves as a reference modality and does not restrict the LLM to select strategies strictly from the dictionary. However, in practice, the LLM tends to exhibit a "lazy" behavior, demonstrating an over-reliance on the provided compression strategies rather than exploring its generative potential to discover optimal methods. This observation reveals an interesting trade-off: while the dictionary module improves stability by providing structured guidance, it may implicitly bias the LLM toward predefined strategies, reducing its incentive to explore more adaptive compression policies.

## C Training and Inference Cost

We provide a systematic comparison of the training and inference costs between our prompt-based method and baseline approaches. The evaluated metrics include model training time, memory usage, average compression time per sample, and average reconstruction time per sample. Detailed results are reported in Table 11. In addition, we use Gemini-3.1 Pro and Gemma4-31B as representative examples to analyze token usage, which is measured by the Google AI Studio platform[6]. For the other LLMs, precise input and output token counts are not reported due to the limitations of their website. Since all prompt-based experiments in this work are conducted through web-based APIs rather than local deployment, we do not report memory consumption for LLMs.

The results show that, while LLM-based methods eliminate training costs, they introduce higher inference latency compared with traditional baselines. By comparing the inference time across different models, we observe that lightweight models do not necessarily lead to shorter inference time. This may be attributed to multiple factors. On the one hand, latency measured through web-based APIs is affected by hardware configurations, server-side scheduling, network conditions, and other deployment-related factors. On the other hand, for complex tasks such as WCR, smaller models may have weaker capabilities in data understanding and structured generation, leading to less stable, more verbose, or self-corrective outputs. This can further increase the overall inference time and output token consumption.

The token-usage comparison in Table 13 further shows that LLM-based inference requires substantial token costs regardless of model scale. Moreover, the larger model, Gemini-3.1 Pro, produces more output tokens and key prompt tokens. This is because its stronger task-processing capability enables it to capture and describe more latent structural features of wireless signals, resulting in richer intermediate analyses and more detailed outputs.

We also conduct experiments to investigate communication overhead under two configurations: using the lowest reasoning depth and a minimal prompt that retains only the task description and data. The results are shown in Table 12. We find that, under both configurations, our proposed method still outperforms the baselines, while the as-

[6]Google AI Studio: https://aistudio.google.com/prompts/new_chat

| METHOD | TRAINING TIME (S) | MEMORY USAGE (MB) | COMPRESSION TIME (S) | RECONSTRUCTION TIME (S) | AVG. NMSE (DB) |
|---|---|---|---|---|---|
| MuSAC-C | 121.79 | 54.9 | $< 0.1$ | $< 0.1$ | -4.71 |
| LSTM | 58.27 | 110.3 | $< 0.1$ | $< 0.1$ | -4.96 |
| TCN | 347.44 | 69.1 | $< 0.1$ | $< 0.1$ | -5.68 |
| CsiNet | 115.83 | 108.7 | $< 0.1$ | $< 0.1$ | -7.19 |
| Fine-tuning LLM | 2873.95 | 2926.6 | $< 0.1$ | $< 0.1$ | -9.29 |
| Mathematical methods | 0 | 0 | $< 0.1$ | $< 0.1$ | -11.30 |
| Qwen3.6-27B | 0 | N/A | 96.3 | 297.8 | -11.00 |
| Gemma4-31B | 0 | N/A | 173.4 | 315.5 | -11.34 |
| Qwen3.5-397B-A17B | 0 | N/A | 123.5 | 293.5 | -17.75 |
| Gemini-3.1 Pro | 0 | N/A | 65.4 | 110.9 | -20.70 |
| GPT-5.4 Thinking | 0 | N/A | 58.6 | 52.4 | -22.03 |

Table 11: Comparison of training and inference cost between our prompt-based method and baseline approaches

| METHOD | COMPRESSION TIME (S) | RECONSTRUCTION TIME (S) | TOTAL TOKENS | AVG. NMSE (DB) |
|---|---|---|---|---|
| Deep Learning models | $< 0.1$ | $< 0.1$ | – | -5.63 |
| Fine-tuning LLM | $< 0.1$ | $< 0.1$ | – | -9.29 |
| Mathematical methods | $< 0.1$ | $< 0.1$ | – | -11.30 |
| **Gemini-3.1 Pro (Minimal prompt + Low Thinking mode)** | **12.2** | **30.8** | **7230** | **-14.86** |
| Gemini-3.1 Pro | 65.4 | 110.9 | 39369 | -20.33 |
| **GPT-5.5 (Minimal prompt + Instant mode)** | **2.9** | **7.6** | **N/A** | **-17.02** |
| GPT-5.4 Thinking | 58.6 | 52.4 | N/A | -22.03 |

Table 12: Comparison of inference overhead and reconstruction performance using minimal prompts and reduced reasoning depth.

| METHOD | INPUT TOKENS | OUTPUT TOKENS | KEY PROMPT TOKENS |
|---|---|---|---|
| Gemini-3.1 Pro | 8142 | 31227 | 319 |
| Gemma4-31B | 7980 | 21318 | 166 |

Table 13: Token usage comparison across different models.

sociated overhead is substantially reduced to a level acceptable for deployment in real-world systems with less stringent real-time requirements. Meanwhile, since we access LLMs via a Web API, the measured runtime also includes additional overhead such as communication latency and queuing time, and may further depend on the server hardware configuration. If the model is deployed locally in future applications, these additional overheads could be further reduced.

Moreover, representing CBRs in textual form leads to substantial token overhead due to the high-dimensional structure of the raw signal data. This reflects an inherent trade-off between semantic flexibility and computational efficiency. We explicitly acknowledge this issue in the Limitations section and identify reducing inference latency and token costs as an important direction for future work.

## D Prompt templates

In this section, we provide the full details of the physics-informed prompt in Figure 6,7,8,9.

## Compression Prompt in Static Scenario

**<Role>:** You are expert in wireless communication.

**<Dataset Description>:** We captured the HE Compressed Beamforming Report from the Action No ACK frame in 802.11 AX communication, which contains Feedback matrices. These matrices are essentially compressed representations of the CSI V matrix after quantization.

**<Hardware Configuration>:**
- Channel frequency: 5785 MHz
- Bandwidth: 20 MHz
- Router Transmit Antennas (Tx): 4
- Mobile Phone Receive Antennas (Rx): 2
- MU-MIMO: Enabled

**<Scenario Information>:**
Current scenario is a "Static scenario"
In the main communication path between the mobile phone and the router, there is no disturbance to the signal and no abrupt changes in the physical scenario. The overall scenario is in a **static and stable state**.

**<Scenario-specific knowledge>:**
Therefore, in stable scenario, Current feedback matrices will exhibit:
Static scenario often leads to smoother signal variations in both temporal and frequency domain.

**<Data Format Description:>:**
- The original feedback matrix is 64 rows × 10 columns.
- Column names: φ11, φ21, φ31, ψ21, ψ31, ψ41, φ22, φ32, ψ32, ψ42.
- Row Mapping: Each row corresponds to a single OFDM subcarrier.
- φ Range: φ ∈ [0, 63] (modulo 64 wrap-around).
- ψ Range: ψ ∈ [0, 15] (modulo 16 wrap-around).
- Processing Requirements: Must handle quantization jumps and wrap-around effects for both φ and ψ (e.g., φ = -1 ≈ φ = 63; ψ = 16 ≈ ψ = 0).

**<Task Instruction>:**
Please perform the following tasks in sequence and use your chain of thought to deduce the optimal strategy
(1) Compress the Current feedback matrices to **{compress ratio}** of their original dimensions
- The dimensions of the output matrix must not exceed **{compress ratio}** of the original dimensions.
- Multi-stage compression is permitted (e.g., combining different compression methods).

(2) Automatically evaluate the availability of Previous feedback matrices
During your reasoning process, you may:
- Compare multiple reversible compression strategies.
- Leverage scenario stability (e.g., consistent trends, differential stability).
- Determine whether to reference the Previous matrices.
- Autonomously select the optimal scheme (Objective: lowest NMSE).

(3) Ensure the compressed data can be reconstructed with low NMSE by subsequent LLM
- The output must be in a strict matrix format.
- Crucial reconstruction hints must be included in the Key prompt.
- If the Previous feedback matrices are utilized, this must be explicitly indicated (assume the reconstructor already possesses the Previous feedback matrices by default).

(4) The output format must strictly adhere to the following structure (no deviation allowed)
Compressed matrices: {Compressed matrices (strict matrix format)}
Key prompt: {Key hints (indicate whether previous feedback matrices were referenced)}

<**Dictionary Module>:**
- Analyze the inter-row trends, inter-column stationarity, and the stability between adjacent time frames within the matrices.
- Evaluate the impact of scenario stability on differential/trend compression.
- The following four methods are the reference methods:

(1) Differential Quadratic Polynomial Fitting (Applicable to static scenarios with a 1/32 compression ratio)
- Unlike traditional quadratic polynomial fitting, this method references the Previous feedback matrices. Because the scenario is stable and inter-frame changes are smooth, modeling the differences significantly reduces the amount of information to be transmitted, achieving a final compression ratio of 1/32.
- Calculate the signed differences with modulus consideration (accounting for quantization wrap-around): use modulo 64 differences for φ columns and modulo 16 differences for ψ columns. Adjust the differences to the minimum absolute value representation (e.g., if difference = (a-b) (mod M) > M/2 , subtract M to get the signed difference).
- Observing that the differences change smoothly across subcarriers (row indices) with obvious trends (dominated by low-frequency components), apply quadratic polynomial fitting to the columns with the most significant changes. Calculate the variance of the differences for each column and select the 6 columns with the highest variance for modeling (the order of their labels must be listed in the Key prompt).
- For each selected column, use least-squares to fit a quadratic polynomial d(r) ≈ a*r^2 + b*r + c (where r is the row index from 0..63). Store the three coefficients for each column (18 coefficients in total). For unselected columns, the reconstruction directly uses the values from the previous frame (assuming no change, or that the overall trend captured by the quadratically fitted columns is sufficient to ignore independent modeling for these columns).
- At the Reconstructor end, reconstruct the predicted difference for each row based on the coefficients according to the method specified in the Key prompt, round to integers, and add them back to the previous frame using modulo 64 or modulo 16 respectively to obtain the reconstructed matrix.

(2) Differential First-Order Fitting (Applicable to static scenarios with a 1/32 compression ratio)
- Adopts a "differential + first-order linear fitting (on the difference sequence)" strategy: For each column, first use the Previous frame as a baseline to calculate the "phase difference sequence" between the Current and Previous frames. Perform a first-order least-squares fit on this difference sequence to obtain the intercept (constant term) and slope (increment per row).
- Only transmit two parameters (intercept, slope) per column. The decoder recovers the full 64 rows of data based on the previous frame + predicted differences. This strategy leverages scenario and inter-frame stability (small, smooth differences) to drastically reduce the required transmission information.

(3) Anchor Point Method (Applicable to various compression ratios and scenarios)
- Select x row positions as anchor points, where x is determined by the compression ratio. The selected rows should ideally represent smooth variations and single jumps (e.g., quantization wrap-around) clearly, or be the first and last rows to indicate the overall trend of the subcarrier data, or be sampled uniformly at equal intervals.
- Explicitly define the reconstruction algorithm in the Key prompt: wrap-around interpolation using the shortest arc (modulo 64 for the φ domain [0–63]; modulo 16 for the ψ domain [0–15]). Autonomously determine whether to use the previous frame for reconstruction assistance based on scenario stability.

(4) Quadratic Polynomial Fitting Method (Applicable to various compression ratios and scenarios)
- Because angles are quantized and subject to wrap-around (modulo) issues, directly applying polynomial fitting to integers will cause destructive jumps at the wrap-around points. Therefore, first perform "phase unwrapping" to continuous real values (in radians). Map each column to radians according to its period: value_rad = value_quant * (2π / period), where period = 64 (for φ columns) or 16 (for ψ columns).
- Perform least-squares fitting on the unwrapped radian values against the subcarrier index i (0..63) : value_rad(i) ≈ a*i² + b*i + c, to obtain and store the coefficients a, b, c (in radians).
- During reconstruction, round v_rad / (2π/period) to obtain the quantized integer, then apply modulo over the period (e.g., val = round(v_rad/(2π/period)) mod period).
- Depending on the specific compression ratio, more key parameters can be embedded in the Compressed matrices and explained in the Key prompt.

<**Input Data>:**
Current feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]
Previous feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

## Output

**Compressed matrices**: [[xx,xx,.....],.....,[xx,xx,.....]]

<**Key prompt>**: {A description of the compression method along with recovery instructions}

Figure 6: Compression prompt in static scenario.

## Compression Prompt in Dynamic Scenario

**<Role>:** You are expert in wireless communication.

**<Dataset Description>:** We captured the HE Compressed Beamforming Report from the Action No ACK frame in 802.11 AX communication, which contains Feedback matrices. These matrices are essentially compressed representations of the CSI V matrix after quantization.

**<Hardware Configuration>:**
- Channel frequency: 5785 MHz
- Bandwidth: 20 MHz
- Router Transmit Antennas (Tx): 4
- Mobile Phone Receive Antennas (Rx): 2
- MU-MIMO: Enabled

**<Scenario Information>:**
Current scenario is a "Dynamic scenario"

In the main communication path between the mobile phone and the router, a man is walking, causing disturbance to the signal. The overall scenario is in a **dynamic and unstable state**.

**<Scenario-specific knowledge>:**
Therefore, in Dynamic scenario, Current feedback matrices will exhibit:

- Stronger fluctuations compared to previous feedback matrices.
- Highly likely to cause rapid phase transitions.
- Decrease inter-carrier coherence.
- Cause instantaneous changes in multipath structures.

You need to model and handle the more drastic changes brought about by the dynamic environment.

**<Data Format Description:>:**
- The original feedback matrix is 64 rows × 10 columns.
- Column names: φ11, φ21, φ31, ψ21, ψ31, ψ41, φ22, φ32, ψ32, ψ42.
- Row Mapping: Each row corresponds to a single OFDM subcarrier.
- φ Range: φ ∈ [0, 63] (modulo 64 wrap-around).
- ψ Range: ψ ∈ [0, 15] (modulo 16 wrap-around).
- Processing Requirements: Must handle quantization jumps and wrap-around effects for both φ and ψ (e.g., φ = -1 ≈ φ = 63; ψ = 16 ≈ ψ = 0).

**<Task Instruction>:**
Please perform the following tasks in sequence and use your chain of thought to deduce the optimal strategy

(1) Compress the Current feedback matrices to **{compress ratio}** of their original dimensions
- The dimensions of the output matrix must not exceed **{compress ratio}** of the original dimensions.
- In dynamic scenarios, you must account for increased volatility, trend modeling/differential features are no longer as smooth as in static scenarios.
- Multi-stage compression is permitted (e.g., combining different compression methods).

(2) Automatically evaluate the availability of Previous feedback matrices
- If dynamic changes are significant, the Previous feedback matrices can only provide partial assistance.
- You must autonomously determine whether to utilize the previous frame.
- If used, it must be explicitly stated in the Key prompt.
- Independent encoding or robust compression strategies should be applied to segments experiencing abrupt changes.

(3) Ensure the compressed data can be reconstructed with low NMSE by subsequent LLM
- The output must be in a strict matrix format.
- Crucial reconstruction hints must be included in the Key prompt.
- For rapidly changing dimensions in dynamic scenarios, you must provide hints on how to perform the reconstruction.

(4) The output format must strictly adhere to the following structure (no deviation allowed)

Compressed matrices: {Compressed matrices (strict matrix format)}

Key prompt: {Key hints (indicate whether previous feedback matrices were referenced)}

**<Dictionary Module>:**
- Evaluate the impact of the dynamic scenario on matrix fluctuations.
- Analyze the differential stability and abrupt changes (discontinuities) between the Current feedback matrices and the Previous feedback matrices.
- The following four methods are the reference methods:

(1) Anchor Point Method (Applicable to various compression ratios and scenarios)
- Select x row positions as anchor points, where x is determined by the compression ratio. The selected rows should ideally represent smooth variations and single jumps (e.g., quantization wrap-around) clearly, or be the first and last rows to indicate the overall trend of the subcarrier data, or be sampled uniformly at equal intervals.
- Explicitly define the reconstruction algorithm in the Key prompt: wrap-around interpolation using the shortest arc (modulo 64 for the φ domain [0–63]; modulo 16 for the ψ domain [0–15]). Autonomously determine whether to use the previous frame for reconstruction assistance based on scenario stability.

(2) Quadratic Polynomial Fitting Method (Applicable to various compression ratios and scenarios)
- Because angles are quantized and subject to wrap-around (modulo) issues, directly applying polynomial fitting to integers will cause destructive jumps at the wrap-around points. Therefore, first perform "phase unwrapping" to continuous real values (in radians). Map each column to radians according to its period: value_rad = value_quant * (2π / period), where period = 64 (for φ columns) or 16 (for ψ columns).
- Perform least-squares fitting on the unwrapped radian values against the subcarrier index i (0..63) : value_rad(i) ≈ a*i² + b*i + c, to obtain and store the coefficients a, b, c (in radians).
- During reconstruction, round v_rad / (2π/period) to obtain the quantized integer, then apply modulo over the period (e.g., val = round(v_rad/(2π/period)) mod period).
- Depending on the specific compression ratio, more key parameters can be embedded in the Compressed matrices and explained in the Key prompt.

(3) Cubic Polynomial Fitting Method (Applicable to 1/16 compression ratio and various scenarios)
- Phase Unwrapping: Before fitting, apply wrap-around correction to each column based on its modulus (φ → 64, ψ → 16). Use a cumulative correction approach: "if the difference between adjacent values exceeds modulus/2, add or subtract the entire modulus" to obtain the shortest-path continuous sequence (avoiding the impact of multi-value jumps on the fitting).
- Polynomial Fitting: Perform least-squares cubic polynomial fitting on each unwrapped column (row index r = 0..63) to obtain coefficients a3,a2,a1,a0 (real numbers). These coefficients become the values in the compressed matrix (the four rows correspond to a3→a0 respectively).
- Storage: Directly store the coefficients (floating-point numbers), and specify the column order, coefficient order, and modulus information in the Key prompt to facilitate subsequent recovery by Large Language Models.

(4) Sparse Method (Applicable to low compression ratios and various scenarios)
Based on the provided Feedback Matrices and the previous frame's data, utilize inter-frame prediction and sparse representation for compression. Assuming the current frame has minor differences from the previous frame, only store the row indices, column indices, and difference values of the changed points. During reconstruction, use the previous frame data and apply modulo arithmetic on the difference values to recover the current frame (modulo 64 for φ columns, modulo 16 for ψ columns).

- Prioritize reversibility and robustness in dynamic environments.

**<Input Data>:**

Current feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

Previous feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

## Output

**Compressed matrices**: [[xx,xx,.....],.....,[xx,xx,.....]]

**<Key prompt>**: {A description of the compression method along with recovery instructions}

Figure 7: Compression prompt in dynamic scenario.

## Reconstruction Prompt in Static Scenario

**<Role>:** You are expert in wireless communication.

**<Dataset Description>, <Hardware Configuration>:**

Using a wireless network packet capture device, we intercepted Action No ACK frames during the communication between a smartphone and a router based on the 802.11ax protocol. The HE Compressed Beamforming Report contained within these frames is essentially:

- The V matrix resulting from the SVD (Singular Value Decomposition) of the CSI (Channel State Information).
- To reduce feedback overhead, the receiving end performed protocol-level compression on the V matrix to form the Feedback matrices.
- The transmitting device subsequently used LLM to further compress the Feedback matrices.

**<Scenario Information>:**
Current scenario is a "Static scenario"
In the main communication path between the mobile phone and the router, there is no disturbance to the signal and no abrupt changes in the physical scenario. The overall scenario is in a **static and stable state**.

**<Scenario-specific knowledge>:**
Therefore, in stable scenario, Current feedback matrices will exhibit:
Static scenario often leads to smoother signal variations in both temporal and frequency domain.

**<Data Format Description:>:**

- The original matrix is 64 rows × 10 columns.
- The column names must strictly be: φ11, φ21, φ31, ψ21, ψ31, ψ41, φ22, φ32, ψ32, ψ42.
- Value changes φ: 0–63.
- Value changes ψ: 0–15.
- Quantization wrapping must be considered to avoid erroneous jumps.
- The results should maintain smooth continuity with the Previous feedback matrices, because no physical variations occur in the scenario.
- The output must be physically interpretable.

**<Task Instruction>:**

Please leverage your reasoning capabilities to achieve the following objectives:

- Reconstruct the original 64*10 feedback matrices from the compressed matrices.
- Synthesize the Key prompt, quantization rules, subcarrier structure, temporal differences and scenario stationarity to minimize the NMSE.
- If the method description in the Key prompt includes a specific decoding approach, please synthesize it but do not be rigidly bound by it.

Output Requirements:

1. Provide the complete reconstructed matrix (64×10) in the exact same format as the Compressed matrices.

2. Write the matrix into a .json file.

3. The output format is as follows:

Reconstructed Matrices:{

(Final reconstructed matrix, 64×10)

}

<**Input Data>:**

Compressed matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

Previous feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

**<Key Prompt>: .....**

## Output

**Reconstructed Matrices**: [[xx,xx,.....],.....,[xx,xx,.....]]

Figure 8: Reconstruction prompt in static scenario.

## Reconstruction Prompt in Dynamic Scenario

**<Role>:** You are expert in wireless communication.

**<Dataset Description>, <Hardware Configuration>:**

Using a wireless network packet capture device, we intercepted Action No ACK frames during the communication between a smartphone and a router based on the 802.11ax protocol. The HE Compressed Beamforming Report contained within these frames is essentially:

- The V matrix resulting from the SVD (Singular Value Decomposition) of the CSI (Channel State Information).
- To reduce feedback overhead, the receiving end performed protocol-level compression on the V matrix to form the Feedback matrices.
- The transmitting device subsequently used LLM to further compress the Feedback matrices.

**<Scenario Information>:**
Current scenario is a "Dynamic scenario"

In the main communication path between the mobile phone and the router, a man is walking, causing disturbance to the signal. The overall scenario is in a **dynamic and unstable state**.

**<Scenario-specific knowledge>:**
Therefore, in Dynamic scenario, Current feedback matrices will exhibit:

- Stronger fluctuations compared to previous feedback matrices.
- Highly likely to cause rapid phase transitions.
- Decrease inter-carrier coherence.
- Cause instantaneous changes in multipath structures.

**<Data Format Description:>:**
- The original matrix is 64 rows × 10 columns.
- The column names must strictly be: φ11, φ21, φ31, ψ21, ψ31, ψ41, φ22, φ32, ψ32, ψ42.
- Value changes φ: 0–63.
- Value changes ψ: 0–15.
- You must account for quantization wraparound, periodicity, and non-stationary offsets caused by dynamic perturbations.
- The output must be physically interpretable.

**<Task Instruction>:**

Please leverage your reasoning capabilities to achieve the following objectives:

- Reconstruct the original 64*10 feedback matrices from the compressed matrices.
- Infer the channel variation patterns to execute non-stationary reconstruction within dynamic scenarios.
- Synthesize the Key prompt, quantization rules, subcarrier structure, and temporal differences to minimize the NMSE.
- Construct an adaptive reconstruction strategy based on the compression method and the dynamic perturbation characteristics (you do not need to strictly adhere to the Key prompt).

To achieve the minimum NMSE, you can leverage the chain-of-thought reasoning to adopt the following strategies:

- Dynamic Differential Compensation: Use the Previous matrices as a baseline to estimate the gradient of change, rather than relying on smooth approximation.
- Multipath Perturbation Modeling: Infer the frequency-selective fading caused by walking based on the correlation between subcarriers.
- Quantization Wraparound Correction: Perform periodic mapping on φ (0–63) and ψ (0–15) to avoid jump artifacts.
- Reverse Compression Reconstruction: Utilize the structural hints provided in the Key prompt to construct a reversible mapping.
- Subcarrier Time-Frequency Consistency: Maintain physical rationality (e.g., subcarrier continuity, preventing unreasonable abrupt changes in adjacent subcarriers).

Output Requirements:

1. Provide the complete reconstructed matrix (64×10) in the exact same format as the Compressed matrices.

2. Write the matrix into a .json file.

3. The output format is as follows:

Reconstructed Matrices:{

(Final reconstructed matrix, 64×10)

}

<**Input Data>:**

Compressed matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

Previous feedback matrices: [[xx,xx,.....],.....,[xx,xx,.....]]

**<Key Prompt>: .....**

## Output

**Reconstructed Matrices**: [[xx,xx,.....],.....,[xx,xx,.....]]

Figure 9: Reconstruction prompt in dynamic scenario.