

# Leveraging Large Language Models for Wireless Symbol Detection via In-Context Learning

Momin Abbas      Koushik Kar      Tianyi Chen  
Department of Electrical, Computer and Systems Engineering  
Rensselaer Polytechnic Institute, Troy, NY, USA

**Abstract**—Deep neural networks (DNNs) have made significant strides in tackling challenging tasks in wireless systems, especially when an accurate wireless model is not available. However, when available data is limited, traditional DNNs often yield subpar results due to underfitting. At the same time, large language models (LLMs) exemplified by GPT-3, have remarkably showcased their capabilities across a broad range of natural language processing tasks.

**But whether and how LLMs can benefit challenging non-language tasks in wireless systems is unexplored.**

In this work, we propose to leverage the *in-context learning* ability (a.k.a. prompting) of LLMs to solve wireless tasks in the low data regime without any training or fine-tuning, unlike DNNs which require training. We further demonstrate that the performance of LLMs varies significantly when employed with different prompt templates. To solve this issue, we employ the latest LLM calibration methods. Our results reveal that using LLMs via ICL methods generally outperforms traditional DNNs on the symbol demodulation task and yields highly confident predictions when coupled with calibration techniques.

**Index Terms**—Large language models, in-context learning, uncertainty quantification, wireless, symbol detection.

## I. INTRODUCTION

### A. Context and Motivation

As the era of AI unfolds, it is expected that deep learning models will play a central role in shaping the future of wireless systems [1]. Most work on AI in wireless communication leverages deep neural networks (DNNs) [2, 3, 4, 5]. To successfully integrate deep learning models into wireless systems, a key requirement is the ability to rapidly adapt to changing environmental conditions, even with limited information about the wireless systems [6, 7]. This includes their ability to handle constantly changing wireless channel conditions using only a few pilot signals [8].

DNN-based nonlinear channel predictors have been proposed through training of recurrent neural networks [9], convolutional neural networks [10], and multi-layer perceptrons [11]. However, several studies, including [11, 12], have reported that deep learning based predictors tend to require

a large number of training data, while failing to outperform well-designed linear filters in the low-data regime. This challenge becomes pronounced as neural networks increase in depth; see Table IV. This is critical in resource-constrained wireless systems, where the acquisition of data is expensive, necessitating costly hardware and skilled labor.

At the same time, Large Language Models (LLMs) have been effectively utilized to address a variety of Natural Language Processing (NLP) and Computer Vision (CV) tasks [13, 14]. Despite significant advancements in NLP and CV, pre-trained LLMs have faced limitations in their development within non-linguistic tasks, let alone wireless tasks. Therefore, combining wireless communications and natural language remains a challenge to utilize these capabilities.

In this work, we strive to achieve the best of both worlds by leveraging *in-context learning* abilities of LLMs on the wireless symbol demodulation task. Our contributions are summarized below:

- C1) We highlight the challenge in training traditional DNNs for symbol demodulation with limited data. To overcome this, we propose harnessing the in-context learning (ICL) abilities of LLMs through inference alone, without requiring any subsequent training or fine-tuning.
- C2) As LLMs via ICL for wireless data is sensitive to changes in prompt templates, we propose employing state-of-the-art (SOTA) calibration methods [15, 16] designed for LLMs.
- C3) We empirically show that ICL methods generally outperform traditional DNNs in scenarios with limited data (e.g. 22% performance improvement for 32-shots).

### B. Related Work

The majority of research in AI for communications relies on traditional frequentist learning methods that use traditional DNNs [2, 3, 4]. These methods involve minimizing the (regularized) training loss, which serves as an estimate of the ground-truth population loss. However, in scenarios with limited data, this estimate becomes unreliable. Consequently, focusing on a single, optimized model parameter vector often results in inaccurate and poorly calibrated probabilistic predictors, leading to overconfident decisions [17, 18].

The work of M. Abbas and T. Chen was supported by NSF CAREER 2047177, NSF ECCS 2412486, Cisco Research Award, and the IBM through the IBM-Rensselaer Future of Computing Research Collaboration.

Some methods focus on enhancing the calibration of DNNs through a validation-based post-processing phase. Platt scaling and temperature scaling [17, 19] determine a fixed parametric mapping of the trained model output that minimizes the validation loss. In contrast, isotonic regression [20] utilizes a non-parametric binning approach. However, since these models primarily target either simple machine learning models or traditional DNNs, they often perform poorly in scenarios with limited data. [21, 22] examine how conformal prediction can be utilized as a general framework to ensure that AI models provide decisions with formal calibration guarantees. However, their notion of calibration differs significantly from ours. They transform probabilistic predictors into set predictors, where the set predictor is considered well-calibrated if it contains the correct output, and their goal does not prioritize performance improvement.

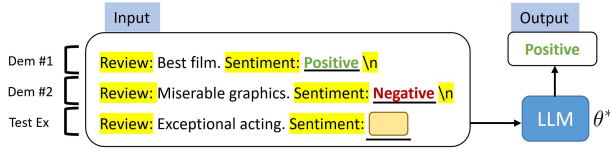


Fig. 1: Example of ICL with LLM  $\theta^*$ .

Recently, LLMs such as GPT-3 [14] have showcased the capability of in-context learning. This feature enables a model to generate suitable outputs for a given query input by leveraging a prompt containing input-output example pairs tailored to the task at hand. ICL has proven to be highly effective in linguistic tasks with limited data, as it operates without the need for explicit training. However, ICL performance fluctuates across various prompt templates due to inadequate calibration [15]. In our work, we employ advanced LLM calibration methods, as proposed in [15, 16], to achieve high performance on wireless data while ensuring precise calibration of the LLMs.

## II. FORMULATION AND SOLUTION APPROACH

In this section, we introduce the data model and then explore the difference between DNNs and LLMs.

### A. Wireless Symbol Demodulation

We consider the wireless symbol demodulation problem from a discrete constellation, relying on received baseband signals that are susceptible to hardware imperfections, noise, and fading [18, 23, 24]. Define  $y_i$  as the  $i$ -th transmitted symbol, and  $x_i$  as the corresponding received signal. Each transmitted symbol  $y_i$  is drawn uniformly at random from a given constellation  $\mathcal{Y}$ . We model  $I/Q$  imbalance at the transmitter and phase fading as in [21]. Accordingly, the ground-truth channel law connecting symbols  $y_i$  into received samples  $x_i$  is described by the equality

$$x_i = e^{j\Psi} f_{IQ}(y_i) + v_i, \quad (1)$$

for some random phase  $\Psi \sim U[0, 2\pi)$ ,  $v_i \sim \mathcal{CN}(0, \text{SNR}^{-1})$  and  $I/Q$  imbalance function  $f_{IQ}(y_i) = \bar{y}_i^I + j\bar{y}_i^Q$  [25] where

$$\begin{bmatrix} \bar{y}_i^I \\ \bar{y}_i^Q \end{bmatrix} = \begin{bmatrix} 1 + \epsilon & 0 \\ 0 & 1 - \epsilon \end{bmatrix} \begin{bmatrix} \cos \delta & -\sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} \begin{bmatrix} y_i^I \\ y_i^Q \end{bmatrix} \quad (2)$$

where  $y_i^I$  and  $y_i^Q$  denote the real and imaginary parts of the modulated symbol  $y_i$ , and  $\bar{y}_i^I$  and  $\bar{y}_i^Q$  represent the real and imaginary parts of the transmitted symbol  $f_{IQ}(y_i)$ . In (2), the channel state is defined by the tuple  $(\Psi, \epsilon, \delta)$ , encompassing the complex phase  $\Psi$  and the  $I/Q$  imbalance  $(\epsilon, \delta)$ .

### B. Deep Neural Networks (DNNs)

We consider a supervised learning setup with a dataset  $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ , consisting of  $N$  examples represented as inputs  $x_i$  corresponding outputs  $y_i$ . The goal is to make predictions for new, unseen test inputs  $x_{\text{test}}$  with an unknown output  $y_{\text{test}}$ .

We are given a probabilistic predictor that implements a parametric conditional distribution model  $p(y_{\text{test}}|x_{\text{test}}, \phi)$  on the output  $y_{\text{test}} \in \mathcal{Y}$  from input  $x_{\text{test}} \in \mathcal{X}$ , where  $\phi \in \Phi$  denotes parameters of a DNN model. Given the training data set  $\mathcal{D}$ , the training algorithm produces an optimized  $\phi_{\mathcal{D}}^*$ . For example, for a classification problem with  $K$  labels (i.e.  $|\mathcal{Y}| = K$ ),  $p(y_{\text{test}}|x_{\text{test}}, \phi_{\mathcal{D}}^*) \in \mathbb{R}^K$  represents the last layer post-softmax probability vector. We can then obtain a point prediction  $\hat{y}_{\text{test}}$  for output  $y_{\text{test}}$  given input  $x_{\text{test}}$  as the probability-maximizing output as

$$\hat{y}_{\text{test}}(x_{\text{test}}|\mathcal{D}) = \arg \max_{y_{\text{test}} \in \mathcal{Y}} p(y_{\text{test}}|x_{\text{test}}, \phi_{\mathcal{D}}^*). \quad (3)$$

However, this represents the conventional approach that *uses training data to train a neural network* and then uses the trained model to make predictions on new test instances.

### C. Proposed Approach: LLM-based ICL (LMIC)

The most common method to leverage capabilities of LLMs is to fine-tune the LLM for specific tasks. However, fine-tuning LLMs can be problematic due to instability [26] caused by various hyperparameter configurations, leading to failed runs, unstable outcomes, and overfitting [27]. Moreover, fine-tuning such large models can be costly and requires access to extensive data and the architecture and weights of LLMs, which may not be publicly available [28].

Moreover, applying LLMs to non-language wireless tasks may require architecture adjustments, such as modifying input/output layers and loss functions [29]. Therefore, it is natural to ask: *Can we use LLMs for wireless tasks without altering the architecture or loss function?* We explore this question using the *in-context learning* abilities of LLMs to solve wireless tasks. This approach offers a streamlined “no-code machine learning” framework, enabling individuals with limited programming or machine learning expertise to address wireless tasks effortlessly.

To reduce lengthy fine-tuning processes and eliminate the need for accessing or modifying the model, recent

advancements in LLMs, such as GPT-3, have showcased the capability of *in-context learning*. ICL is a *training-free* approach enabling the model to generate appropriate outputs for test samples by using prompts containing task-specific input-output examples. This approach works through an API without requiring direct access to the LLM. A visual representation of ICL is provided in Fig. 1.

Specifically, ICL aims to predict a test sample  $x_{\text{test}}$  by conditioning on a prompt sequence  $(f_x(x_1), f_y(y_1), \dots, f_x(x_N), f_y(y_N), f_x(x_{\text{test}}))$ . This sequence includes  $N$ -shot samples  $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$  (aka demonstrations) and the query test sample  $x_{\text{test}}$ . Here,  $f_x(\cdot)$  and  $f_y(\cdot)$  are template functions that provide predefined text descriptions for input and output, respectively (refer to text highlighted in **yellow** in Fig. 1). Additionally, the output template function  $f_y(\cdot)$  may convert labels  $y_i$  into natural language format instead of numeric/one-hot labels. For instance, in binary classification, it could transform labels  $(0, 1)$  into **(Positive, Negative)** (see labels in Fig. 1). Together,  $f_x(\cdot)$  and  $f_y(\cdot)$  constitute the *prompt template*, providing a textual interpretation of the data. A prompt  $P$  for an input  $x_{\text{test}}$  is defined as:

$$P(x_{\text{test}}, (x_i, y_i)_{i=1}^N) \triangleq d_1 \oplus d_2 \oplus \dots \oplus d_N \oplus f_x(x_{\text{test}}), \quad (4)$$

where each demonstration  $d_i$  is given by  $f_x(x_i) \oplus f_y(y_i)$  and  $\oplus$  denotes the concatenation operation.

Then for a pretrained LLM  $M_{\phi^*}$  parameterized by  $\phi^*$ ,

$$p_{\text{LLM}}(y_{\text{test}}|x_{\text{test}}, \phi^*, \mathcal{D}) \triangleq M_{\phi^*}(P(x_{\text{test}}, (x_i, y_i)_{i=1}^N)). \quad (5)$$

Note that the pre-trained LLM  $M_{\phi^*}$  is *not trained or fine-tuned* on the training data  $\mathcal{D}$  and is therefore independent of  $\mathcal{D}$ . Instead, the training data serves as a ‘context’ within the prompt, comprising a sequence of input-label pairs known as *demonstrations*. Such a capability of an LLM to learn “in-context” presents an intriguing aspect whereby the LLM is capable of acquiring knowledge and performs well on a wide range of downstream tasks without any task-specific fine-tuning [14]. Moreover, for our approach,  $x_i$  ( $y_i$ ) represent the received (transmitted, respectively) signals (see Sec. II-A) and the corresponding prompt is shown in Table I, where the bold numbers indicate the raw data for  $x_i$  ( $y_i$ ).

However, the dimension of  $p_{\text{LLM}}(y_{\text{test}}|x_{\text{test}}, \phi^*, \mathcal{D})$  is not  $K$  as in traditional neural networks; instead, it corresponds to the number of tokens present in the vocabulary of the LLM; this is because LLMs perform next-token prediction, hence the dimension matches the number of tokens in LLM vocabulary. Hence, we proceed by sampling tokens corresponding to the classes in the label space, resulting in a probability vector of size  $K$ . Finally, we can get the prediction following the same rule as Eq. (3), replacing  $p(y'_{\text{test}}|x_{\text{test}}, \phi^*, \mathcal{D})$  by  $p_{\text{LLM}}(y'_{\text{test}}|x_{\text{test}}, \phi^*, \mathcal{D})$ . We use this as our base method and refer to it as *vanilla ICL*.

However, recent studies show that vanilla ICL’s performance varies widely across different prompt templates and

Prompt Template	Label Space
8APSK signals are as follows: Signal 1’s real part is <b>-2</b> and imaginary part is <b>4</b> . Actual Signal: <b>5</b> (since $ Y =8$ ) Test Signal’s real part is <b>3</b> and imaginary part is <b>-1</b> . Actual Signal:	$0, \dots, 7$

TABLE I: The prompts template used for ICL methods; For brevity, here we show only one demonstration.

demonstrations [15, 16], ranging from random guessing to state-of-the-art levels. Additionally, we find that while these GPT-like models perform adequately via vanilla ICL, their predictions lack reliability on wireless tasks when assessed with Shannon entropy (see Figs. 3-4). This observation aligns with the reliability concerns of LLMs identified in linguistic tasks [16]. We gauge this is related to the poor calibration of LLMs [15]. To address these reliability challenges posed by vanilla ICL, we leverage latest state-of-the-art (SOTA) calibration methods for LLMs, namely Contextual Calibration (ConC) [15] and Linear Probe Calibration (LinC) [16]. Accuracy and calibration are independent criteria, with the presence of one not implying the other [21].

For brevity of notation, we denote the output probabilities  $p_{\text{LLM}}(y_{\text{test}}|x_{\text{test}}, \phi^*, \mathcal{D})$  as  $\mathbf{p}$ . Then, the goal is to linearly adjust the output probabilities using an affine transformation, also known as Platt Scaling [19]:

$$\tilde{\mathbf{p}} = \text{softmax}(\mathbf{A}\mathbf{p} + \mathbf{b}), \quad (6)$$

where  $\mathbf{A} \in \mathbb{R}^{K \times K}$  and  $\mathbf{b} \in \mathbb{R}^K$  represent parameters applied to the original probabilities  $\mathbf{p}$  to obtain new probabilities  $\tilde{\mathbf{p}}$ . ConC then uses a prompt  $P_{\text{cf}} = P(\text{“N/A”}, (x_i, y_i)_{i=1}^N)$  (where test point  $x_{\text{test}}$  is replaced with a *content-free* (cf) input such as the string “N/A” and obtain  $\mathbf{p}$  for this content-free input, denoted by  $\mathbf{p}_{\text{cf}}$  i.e.  $\mathbf{p}_{\text{cf}} = M_{\phi^*}(P_{\text{cf}})$ . The parameters are set via (i.e. no training needed)

$$\mathbf{A} = \text{diag}(\mathbf{p}_{\text{cf}})^{-1} \quad \text{and} \quad \mathbf{b} = \mathbf{0}. \quad (7)$$

In contrast, LinC begins with a predefined set of calibration parameters, including zero initialization. Following this, LinC uses a few additional prompts to *train* the matrix  $\mathbf{A}$  and vector  $\mathbf{b}$  before applying the affine transformation (for details, see [16]). However, in our case, we do not use any additional samples and reuse the same  $N$ -shot demonstrations for training low-dimensional parameters  $\mathbf{A}$  and  $\mathbf{b}$ .

### III. NUMERICAL EVALUATION

We apply our LMIC approach, as described in Sec. II-C, to address the symbol demodulation problem [23, 30] in the presence of transmitter hardware imperfections. Unlike previous studies [18, 24], which focused on frequentist and Bayesian learning via traditional DNNs, our goal is to use LMIC to achieve high accuracy and precise calibration under a severely limited resource regime (e.g.,  $< 50$  data samples), where traditional DNNs fail miserably (see Table IV).

Demodulation is implemented via an LLM  $M_{\phi^*}$  as a next-token prediction problem (see Sec. II-C). We used GPT-J [31]

Format#	Prompt Template
1	8APSK signals are as follows: Signal 1's real part is -2 and imaginary part is 4. Actual Signal: 5 Test Signal's real part is 3 and imaginary part is -1. Actual Signal:
2	8APSK signals are as follows: Signal 1's real part is -2 and imaginary part is 4. Actual Constellation: 5 Test Signal's real part is 3 and imaginary part is -1. Actual Constellation:
3	8APSK signals are as follows. Classify the signals based on the true set of classes [0, 1, 2, 3, 4, 5, 6, 7]. Signal 1's real part is -2 and imaginary part is 4. Actual Signal: 5 Test Signal's real part is 3 and imaginary part is -1. Actual Signal:
4	Based on the 8APSK signals shown below, predict the Test Signal's output class from the set of classes [0, 1, 2, 3, 4, 5, 6, 7]: Signal 1's real part is -2 and imaginary part is 4. Actual Signal: 5 Test Signal's real part is 3 and imaginary part is -1. Actual Signal:

TABLE II: A list of different prompt templates that were used to investigate the impact of templates on Llama-2 7B 8-shot setting. For brevity, here we show only one demonstration.

with 6B parameters and two variants of the latest Llama-2 [32] with 7B and 13B parameters<sup>1</sup>.

The last layer implements a softmax classification for the  $K = |Y|$  possible constellation points. We employ the Amplitude-Phase-Shift-Keying (APSK) modulation with  $K = 8$ . The SNR level is set to  $\text{SNR} = 5$  dB. The amplitude and phase imbalance parameters are independent and distributed as  $\epsilon \sim \text{Beta}(\epsilon/0.15|5, 2)$  and  $\delta \sim \text{Beta}(\delta/15^\circ|5, 2)$ , respectively [24]. Unless specified otherwise, LMIC methods employ a fixed prompt template chosen manually to enhance performance, as demonstrated alongside examples in Table I; bold numbers represent the raw data.

All our experiments were conducted on two NVIDIA RTX 3090 GPUs. As mentioned before, we consider the low resource regime, where the number of available training samples is scarce (typically under  $<50$ ). This is especially vital in resource-constrained scenarios where acquiring wireless data is expensive due to the costly hardware and skilled labor.

As baselines, we trained a fully connected deep neural network (DNN) similar to the one considered in [18, 21] with real inputs  $x_i$  of dimension 2, following Eq. (1). It consists of four hidden layers, with 10 neurons in the first hidden layer and 30 neurons in each subsequent hidden layer, each activated by ReLU. We also considered deeper networks with five, six and seven layers with 30 neurons in each additional hidden layer. The final layer performs softmax classification for the  $|Y|$  possible constellation points.

To ensure a fair comparison, each DNN is trained using the identical set of samples employed as demonstrations within the prompt for LMIC methods. For instance, if there are 8-shots (i.e. demonstrations) in the prompt, the DNN baseline is trained using the same set of 8 samples.

Our main results are shown in Table IV. We observe that across most experiments (i.e., 15 out of 21 cases), our LMIC methods, particularly ConC and LinC, consistently demonstrate superior performance compared to the DNN

<sup>1</sup>Note 13B model is the largest that can fit into our current GPU memory.

Model	Vanilla ICL	ConC	LinC
Llama-2 7B	0.2341	0.1166	0.1166

TABLE III: Expected Calibration Error (ECE) comparison between different ICL methods under 32-shot setting.

baselines. This showcases the robust generalization capability of LLMs across various model sizes and few-shot settings. For instance, in the 32-shot experiment, Llama-2 7B outperforms the DNN-4 by a significant margin of about 22% (69.31% vs. 47.52%). Such a capability of ICL to understand contextual information from a handful of samples is particularly intriguing, especially when considering that the data is non-linguistic wireless data. We also note that the Llama-2 model outperforms the GPT-J model. This could be because Llama-2, released in August 2023, is one of the most recent models and therefore pre-trained on larger amounts of more recent data. We also notice that while the performance of DNNs generally declines with an increase in layers, eventually approaching near-guess accuracy with 7 hidden layers, the opposite trend is observed for LLMs: performance improves as the number of parameters increases (c.f. GPT-J 6B vs Llama-2 7/13B). Also, there is no consistent pattern in the performance of varying model sizes within the LLM family (i.e. Llama-2 7B vs 13B), which is consistent with previous works [16]. However, Llama-2 7B notably achieves the highest accuracy of 69.31% for 32-shots. Moreover, we observe that when the number of samples is less than the number of classes (i.e.  $N < K$ ), DNNs with fewer hidden layers usually perform better than our LMIC methods (c.f. GPT-J and Llama-2 13, 5/6-shot results with DNN-4/5). This observation is in line with previous works that emphasize the pivotal role of label space in the success of ICL [33].

As previously mentioned, prior works suggest that the performance of vanilla ICL fluctuates across different prompt templates in linguistic tasks. To investigate if this phenomenon also holds for non-linguistic wireless data, we use ten distinct prompt templates, four of which are listed in Table II (due to space constraints), on Llama-2 7B under 8-shot setting. From Fig. 2, indeed the performance of vanilla ICL is volatile across different prompts while the latest SOTA calibration methods exhibit substantial improvements in accuracy with notably lower variance, highlighting the effectiveness of these methods in enhancing the model's performance across various prompt templates.

We further evaluate the reliability of LLM predictions by employing the *Shannon entropy* metric [34], which measures the expected uncertainty in a probability distribution  $\mathbf{p}$ . A model is considered better when entropy values are lower. For this experiment we used our largest Llama-2 13B model to compare vanilla ICL and LinC, showing results for 4/8/16/32-shots in Figs. 3-4. We observe that employing vanilla ICL results in high entropy values, suggesting that most test predictions were made with very low confidence, indicating

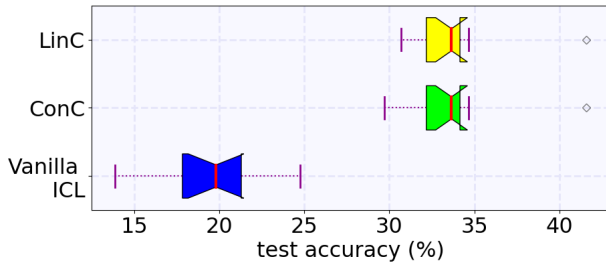


Fig. 2: Comparison across ten different prompt templates.

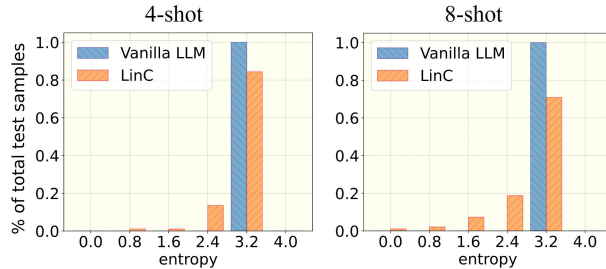


Fig. 3: Shannon entropy histograms of different ICL methods on Llama-2 13B for 4/8-shots setting; we use logarithmic base two.

a tendency towards random guessing. These findings are consistent with prior studies for linguistic data [16]. In contrast, a calibrated LLM via LinC produces significantly lower entropy values, reflecting the increased confidence.

To further evaluate LLM calibration, we utilize the widely-used Expected Calibration Error (ECE) metric [35] to quantify the distance between predicted and actual probabilities. Table III shows the results on Llama-2 7B, 32-shot setting (we only present this setting due to limited space). We observe that LLM calibration methods such as ConC and LinC consistently exhibit much lower ECE when compared to vanilla ICL, highlighting the critical importance of calibrating LLMs for wireless data.

Despite the merits of our approach, it has some limitations. Although our method eliminates the need for GPU-intensive training, it still relies on GPU memory for inference, albeit at a reduced scale compared to training or fine-tuning. Nevertheless, with the increasing adoption of LLMs across various domains, we anticipate that GPUs will become more readily accessible for deployment in wireless communication tasks in the future. Lastly, although we employed manually selected prompts and demonstrations, exploring how to integrate our framework with methods for selecting better demonstrations and prompt templates is an interesting future direction.

#### IV. CONCLUSIONS

While LLMs have been extensively studied for linguistic tasks, their utilization for non-linguistic wireless data remains largely unexplored. In this work, we capitalize on the *in-context learning* abilities of LLMs that not only achieves high performance but also yields highly confident predictions when integrated with SOTA LLM calibration techniques, especially in data-scarce scenarios where traditional DNNs

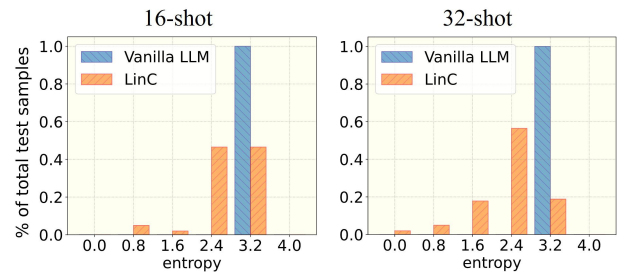


Fig. 4: Shannon entropy histograms of different ICL methods on Llama-2 13B for 16/32-shot setting; we use logarithmic base two.

Model	Shots						
	4	5	6	8	16	24	32
Guessing	12.50	12.50	12.50	12.50	12.50	12.50	12.50
DNN-4	15.84	37.62	33.66	30.69	41.58	31.68	44.55
DNN-5	13.86	28.72	35.64	32.67	39.60	43.56	44.55
DNN-6	15.84	12.87	18.81	23.76	23.76	23.76	37.62
DNN-7	14.85	12.87	12.87	12.87	12.87	12.87	25.74
GPT-J 6B*	19.79	18.81	15.84	16.83	37.62	22.77	18.81
GPT-J 6B <sup>†</sup>	24.75	33.66	27.72	33.66	41.58	47.52	41.58
GPT-J 6B <sup>‡</sup>	24.75	33.66	28.71	35.64	43.56	46.53	41.58
DNN-4	26.73	31.68	41.58	37.62	40.59	50.50	47.52
DNN-5	23.76	24.75	40.59	33.66	39.60	52.48	44.55
DNN-6	21.78	13.86	25.74	20.79	12.87	39.60	24.75
DNN-7	11.88	12.87	12.87	12.87	12.87	22.77	12.87
Llama-7B*	33.33	20.79	26.73	16.83	31.68	58.42	64.36
Llama-7B <sup>†</sup>	29.17	40.59	39.60	41.58	49.50	59.41	69.31
Llama-7B <sup>‡</sup>	29.17	40.59	39.60	41.58	49.50	58.42	69.31
DNN-4	31.68	39.60	36.63	32.67	47.52	59.40	45.54
DNN-5	28.71	22.77	23.76	34.65	32.67	45.54	50.50
DNN-6	22.77	25.74	19.80	21.78	22.77	43.56	30.69
DNN-7	21.78	12.87	17.82	20.79	20.79	12.87	12.87
Llama-13B*	26.04	18.81	31.68	31.25	40.59	54.46	53.47
Llama-13B <sup>†</sup>	37.50	32.67	33.66	38.54	49.50	65.35	58.42
Llama-13B <sup>‡</sup>	37.50	32.67	33.66	38.54	49.50	65.35	58.42

TABLE IV: Performance comparison for the system demodulation task; {} in DNN-{} refers to the number of hidden layers of the fully-connected deep neural network; \* denotes vanilla ICL, <sup>†</sup> denotes ConC, and <sup>‡</sup> denotes LinC.

typically fail. We believe these findings carry important implications for advancing wireless systems through large language models. In the future, our aim is to investigate the high-resource regime, utilizing abundant data and compute to first fine-tune LLMs and then employ our LMIC approach.

#### REFERENCES

- [1] Tugba Erpek, Timothy J O'Shea, Yalin E Sagduyu, Yi Shi, and T Charles Clancy, "Deep learning for wireless communications," *Development and Analysis of Deep Learning Architectures*, pp. 223–266, 2020.
- [2] Osvaldo Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.
- [3] Linglong Dai, Ruicheng Jiao, Fumiyuki Adachi, H Vincent Poor, and Lajos Hanzo, "Deep learning for wireless communications: An emerging interdisciplinary paradigm," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 133–139, 2020.



- [4] Yonina C Eldar, Andrea Goldsmith, Deniz Gündüz, and H Vincent Poor, *Machine learning and wireless communications*, Cambridge University Press, 2022.
- [5] Ruolin Zhou, Fugang Liu, and Christopher W Gravelle, “Deep learning for modulation recognition: A survey with a demonstration,” *IEEE Access*, vol. 8, pp. 67366–67376, 2020.
- [6] Osvaldo Simeone, Sangwoo Park, and Joonhyuk Kang, “From learning to meta-learning: Reduced training overhead and complexity for communication systems,” in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.
- [7] Lisha Chen, Sharu Theresa Jose, Ivana Nikoloska, Sangwoo Park, Tianyi Chen, Osvaldo Simeone, et al., “Learning with limited samples: Meta-learning and applications to communication systems,” *Foundations and Trends® in Signal Processing*, vol. 17, no. 2, pp. 79–208, 2023.
- [8] Tomer Raviv, Sangwoo Park, Osvaldo Simeone, and Nir Shlezinger, “Modular model-based bayesian learning for uncertainty-aware and reliable deep mimo receivers,” in *2023 IEEE ICC Workshops*. IEEE, 2023, pp. 1032–1037.
- [9] Wei Liu, Lie-Liang Yang, and Lajos Hanzo, “Recurrent neural network based narrowband channel prediction,” in *Proc. IEEE 63rd Vehicular Technology Conference*, Melbourne, Australia, 2006, vol. 5, pp. 2173–2177.
- [10] Jide Yuan, Hien Quoc Ngo, and Michail Matthaiou, “Machine learning-based channel prediction in massive mimo with channel aging,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2960–2973, 2020.
- [11] Hwanjin Kim, Suheol Kim, Hyeongtaek Lee, Chulhee Jang, Yongyun Choi, and Junil Choi, “Massive mimo channel prediction: Kalman filtering vs. machine learning,” *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 518–528, 2020.
- [12] Wei Jiang, Mathias Strufe, and Hans Dieter Schotten, “Long-range mimo channel prediction using recurrent neural networks,” in *Proc. IEEE Annual Consumer Communications & Networking Conference*, Las Vegas, NV, 2020, pp. 1–6.
- [13] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon, “Unified language model pre-training for natural language understanding and generation,” in *Advances in Neural Information Processing Systems*, 2019, vol. 32.
- [14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [15] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh, “Calibrate before use: Improving few-shot performance of language models,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 12697–12706.
- [16] Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen, “Enhancing in-context learning via linear probe calibration,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 307–315.
- [17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, “On calibration of modern neural networks,” in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.
- [18] Kfir M. Cohen, Sangwoo Park, Osvaldo Simeone, and Shlomo Shamai, “Bayesian active meta-learning for reliable and efficient ai-based demodulation,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 5366–5380, 2022.
- [19] John Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Adv. Large Margin Classif.*, vol. 10, 06 2000.
- [20] Bianca Zadrozny and Charles Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 694–699.
- [21] Kfir M Cohen, Sangwoo Park, Osvaldo Simeone, and Shlomo Shamai Shitz, “Calibrating ai models for few-shot demodulation via conformal prediction,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [22] Anastasios N Angelopoulos, Stephen Bates, et al., “Conformal prediction: A gentle introduction,” *Foundations and Trends® in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.
- [23] Ahmed G. Helmy, Marco Di Renzo, and Naofal Al-Dhahir, “On the robustness of spatial modulation to i/q imbalance,” *IEEE Communications Letters*, vol. 21, no. 7, pp. 1485–1488, July 2017.
- [24] Sangwoo Park, Hyeryung Jang, Osvaldo Simeone, and Joonhyuk Kang, “Learning to demodulate from few pilots via offline and online meta-learning,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 226–239, 2020.
- [25] Deepaknath Tandur and Marc Moonen, “Joint adaptive compensation of transmitter and receiver iq imbalance under carrier frequency offset in ofdm-based systems,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5246–5252, 2007.
- [26] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow, “On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines,” in *International Conference on Learning Representations*, 2021.
- [27] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang, “Fine-tuning can distort pretrained features and underperform out-of-distribution,” in *International Conference on Learning Representations*, 2022.
- [28] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al., “Opt: Open pre-trained transformer language models,” *arXiv preprint arXiv:2205.01068*, 2022.
- [29] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee, “Lift: Language-interfaced fine-tuning for non-language machine learning tasks,” in *Advances in Neural Information Processing Systems*, 2022, pp. 11763–11784.
- [30] Yi Zhang, Akash Doshi, Rob Liston, Wai-Tian Tan, Xiaoqing Zhu, Jeffrey G. Andrews, and Robert W. Heath, “Deepwiphy: Deep learning-based receiver design and dataset for ieee 802.11ax systems,” *IEEE Transactions on Wireless Communications*, 2021.
- [31] Ben Wang and Aran Komatsuzaki, “Gpt-j-6b: A 6 billion parameter autoregressive language model,” 2021.
- [32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al., “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [33] Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer, “Rethinking the role of demonstrations: What makes in-context learning work?,” *arXiv preprint arXiv:2202.12837*, 2022.
- [34] Claude E Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [35] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht, “Obtaining well calibrated probabilities using bayesian binning,” in *Proc. of AAAI Conference on Artificial Intelligence*, 2015.