A Semi-Blind Receiver for Time-Selective Channels Using Transformer-Based In-Context Learning

Rahul Rajesh Singh[1,2], Ananthanarayanan Chockalingam[2], and Khushboo Mawatwal[1]

[1]Network Modem Team, Samsung R&D India, Bangalore
[2]Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore

***Abstract*—Transformers leverage the self-attention mechanism to model long-range dependencies in sequential data, with decoder-only architectures enabling auto-regressive next-token prediction. When pre-trained at scale, these models exhibit in-context learning (ICL) ability, adapting them to various tasks without retraining. This enables ICL to be useful for sequential symbol detection in wireless receivers by dynamically inferring channel conditions (e.g., fading, noise) from the available input prompt (a window of recently received pilot symbols). However, using ICL for time-selective fading channels requires sending a long sequence of pilots to detect symbols, which reduces the spectral efficiency. To mitigate this, we propose a semi-blind ICL approach that uses a time-varying augmented prompt formed by combining a limited number of pilots with detected symbols (noisy-labels) from transformer output. This facilitates the transformer to adapt its predictions to current channel and signal-to-noise ratio (SNR) conditions, enhancing symbol detection accuracy without compromising spectral efficiency. In addition, generating transformer feedback requires additional forward passes through the model, which increases the training time. We address this by introducing a transfer learning strategy, i.e., model pre-training followed by fine-tuning. We demonstrate through experimentation that the proposed semi-blind symbol detection using the ICL method gives superior bit error rate performance across diverse modulations (BPSK, QPSK, 16-QAM), SNRs, and Doppler frequency shifts compared to conventional symbol detection methods.**

***Index Terms*—Semi-blind receiver, symbol detection, time-selective channels, transformer, in-context learning, GPT.**

I. Introduction

Reliable symbol detection in wireless communication receivers confronts persistent challenges, particularly in time-selective channels where balancing accuracy and spectral efficiency remains a subtle challenge. Conventional symbol detection methods utilize sparse pilot symbols for channel estimation, followed by equalization to mitigate distortion, and ending with symbol demapping to recover transmitted data symbols. Furthermore, any imperfections or irregularities in channel estimation propagate to symbol demapping, manifesting itself as elevated bit error rate (BER) and degraded throughput performance. Time-selective channels further amplify this issue, requiring increased pilot density to track rapidly evolving fading channel coefficients, reducing spectral efficiency and data throughput.

This work was supported in part by the J. C. Bose National Fellowship and the NSF-DST project (International Cooperation Division), Department of Science and Technology, Government of India.

Recent advances in application of machine learning and deep learning (DL) to wireless communication systems have demonstrated promising applications in channel estimation [1], channel prediction [2], communication receiver [3], and end-to-end communication model [4]. Unlike model-based signal processing paradigms, these data-driven DL approaches often function as black-box systems lacking interpretability [5]. Additionally, DL methods typically require substantial computational resources, including extensive diverse training datasets, large model architectures, and prolonged training durations to address non-linearities and imperfections inherent in practical systems. Consequently, while DL techniques may exhibit strong performance on localized datasets, their real-world applicability is limited due to lack of interpretability and insufficient generalization capabilities. All of these greatly compromise the ability of a DL model to generalize to unseen and rapidly varying channel conditions.

In recent times, *transformer* architecture [6] has revolutionized tasks associated with natural language processing (NLP), sequence modeling, and time-series analysis. Central to this revolution is the *self-attention* mechanism, which dynamically weights the relevance of all input elements to capture long-range dependencies and contextual relationships, enabling parallel processing, improved interpretability, and superior modeling of complex sequential patterns. Subsequent studies [7]–[10] demonstrate that large pre-trained decoder-only transformers can perform *in-context learning* (ICL), adapting to any tasks without explicit parameter updates. In ICL, models infer patterns from an input prompt that contains example input-output (I/O) pairs and a query, to generate a response. The authors in [10] further show that ICL enables generalization to functions not seen during training. This transformer-based ICL framework has been extended to wireless communication tasks such as symbol estimation and detection. For estimation, [11] establishes transformers as an optimal ICL estimator for linear channels, [12] addresses non-linear multiple-input-multiple-output (MIMO) channel using ICL, and [13] introduces iterative symbol estimate refinement using decoder's extrinsic information feedback. For detection, [14] explores the idea of using NLP-prompted GPT-J, while [15] develops decision-feedback-inspired ICL. However, existing methods primarily focus on *block-fading* channels, leaving *time-selective* channel adaptation underexplored.

In this work, we address the problem of symbol detection

in uncoded transmissions over time-selective channels using transformer-based ICL. Our goal is to minimize pilot overhead by adopting a semi-blind ICL approach that leverages data symbols alongside limited pilots to enhance spectral efficiency while preserving BER performance. Key contributions include:

- First, we design a decoder-only transformer model by formulating symbol detection task for ICL using a pilot-only time-varying prompt, and defining transformer's I/O relationship along with a modified loss function to train the model. This method is called BASe In-Context Symbol dEtection over Time-Selective channel (BASICSETS).
- We extend BASICSETS by incorporating detected symbols from the transformer output [15], forming a time-varying augmented prompt. This enhanced method is called SEMi-blind IN-contEXT Symbol dEtection over Time-Selective channel (SEMINEXTSETS). In addition, we present a training process based on transfer learning for SEMINEXTSETS that significantly reduces training time.
- To quantify the effect of error propagation on SEMINEXTSETS, we introduce a genie-aided variant (GENIESETS) that lower-bounds BER performance.
- To demonstrate the efficacy of SEMINEXTSETS, we train and evaluate its BER performance for single-pilot transmission and compare it with existing conventional symbol detection methods.

II. System model

A. Signal model for time-selective channel

We consider a single-input-single-output (SISO) system with relative motion between transmitter and receiver [2], resulting in a time-selective fading channel. Let the data symbol at tth time instant be $x(t)$, randomly sampled from a uniformly distributed M-ary constellation set, $\mathcal{X}$. The received signal, $y(t)$, can be written as

$$y(t) = h(t)x(t) + w(t). \tag{1}$$

Here, $h(t)$ denotes the time-varying channel fading coefficient at the tth time instant, statistically modeled as a circularly symmetric complex Gaussian random variable with zero mean and unit variance, i.e., $P_h \sim \mathcal{CN}(0,1)$. In addition, $w(t)$ represents additive white Gaussian noise (AWGN), statistically modeled as zero mean and σ^2 variance, i.e., $P_w \sim \mathcal{CN}(0,\sigma^2)$. A time-selective fading channel is characterized by a maximum Doppler frequency shift defined as [16], [17]

$$f_{\mathrm{D}}^{\max} = \frac{f_{\mathrm{c}} v}{c}, \tag{2}$$

where f_{c} is the carrier frequency, v is the maximum velocity between the transmitter and the receiver, and c is the speed of light. The coherence time of the channel, T_{c}, is related to the maximum Doppler shift as $T_{\mathrm{c}} \propto \frac{1}{f_{\mathrm{D}}^{\max}}$.

We arrange symbol transmission in frames of duration T (i.e., prompt length), with T_{p} pilot symbols followed by $T_{\mathrm{d}} = T - T_{\mathrm{p}}$ data symbols. Note that channel $h(t)$ is temporally correlated if $T \leq T_{\mathrm{c}}$, otherwise uncorrelated. We denote $x_{\mathrm{p}}(t)$ and $x_{\mathrm{d}}(t)$ as the pilot and data signal transmitted at the tth time instant, respectively. Using (1), received pilot signal can be written as

$$y_{\mathrm{p}}(t) = h(t)x_{\mathrm{p}}(t) + w(t),\ \forall t = 1,\ldots,T_{\mathrm{p}}. \tag{3}$$

Similarly, received data signal can be written as

$$y_{\mathrm{d}}(t) = h(t)x_{\mathrm{d}}(t) + w(t),\ \forall t = T_{\mathrm{p}}+1,\ldots,T. \tag{4}$$

We use channel usage efficiency as a measure of spectral efficiency, defined as

$$\eta = 1 - \frac{T_{\mathrm{p}}}{T}. \tag{5}$$

For large pilot length T_{p}, η reduces and affects the achievable data throughput.

B. Conventional symbol detection methods

Traditionally, symbol detection methods often employ steps that rely on certain statistical assumptions or sub-optimal approximations to balance accuracy and complexity. Consequently, these methods may not achieve optimal performance across all channel and signal-to-noise ratio (SNR) conditions. We employ conventional symbol detection methods as benchmark techniques under both perfect and imperfect channel state information (CSI) availability scenarios. Furthermore, for the SISO system, the maximum likelihood (ML) detector is optimal and remains computationally feasible [18], serving as our baseline for detecting symbols.

1) Perfect CSI based ML symbol detection: Assuming perfect knowledge of $h(t)$, ML detection of the tth data symbol is given by

$$\widehat{x}_{\mathrm{d}}(t) = \underset{x_{\mathrm{d}}(t)\in\mathcal{X}}{\arg\min}\, ||h(t)x_{\mathrm{d}}(t) - y_{\mathrm{d}}(t)||^2,\ \forall t = T_{\mathrm{p}}+1,\ldots,T. \tag{6}$$

2) Estimated CSI based ML symbol detection: In this case, $h(t)$ is estimated at the receiver using pilot symbols. A minimum mean square error (MMSE) based channel estimator is optimal [19], but requires knowledge of conditional probability distributions, which is often impractical to obtain. We instead employ a linear MMSE (LMMSE) estimator and, using (3), it is written as [2]

$$\widehat{h}(t) = \frac{y_{\mathrm{p}}(t)|x_{\mathrm{p}}(t)|^2}{x_{\mathrm{p}}(t)(|x_{\mathrm{p}}(t)|^2 + \sigma^2)}. \tag{7}$$

This channel estimate can be used directly or interpolated (e.g., linear, Wiener [20]) to obtain the channel estimate on the data symbols. Using (4), the ML detection then results in

$$\widehat{x}_{\mathrm{d}}(t) = \underset{x_{\mathrm{d}}(t)\in\mathcal{X}}{\arg\min}\, ||\widehat{h}(t)x_{\mathrm{d}}(t) - y_{\mathrm{d}}(t)||^2,\ \forall t = T_{\mathrm{p}}+1,\ldots,T. \tag{8}$$

III. Transformer-based ICL for symbol detection over time-selective Channel

We employ OpenAI's GPT-2 [21] for its open-source availability and zero-shot capabilities. Note that, as shown in [7], [8], new larger models (e.g., GPT-3) may produce superior few-shot performance due to increased capacity and thus can give better results. In our work, we first introduce symbol detection using the BASICSETS method and then extend it to the SEMINEXTSETS method. Our methodology defined in this section is architecture-agnostic and applicable to any decoder-only transformer.

A. Symbol detection using BASICSETS

1) Problem formulation: A BASICSETS symbol detection task is characterized by a tuple $\beta = \{(h(t))_{t=1}^{T}, \sigma^2\}$, following an unknown joint distribution, $P_\beta = P_H P_{\sigma^2}$, where P_H represents the joint distribution of $(h(t))_{t=1}^{T}$ and P_{σ^2} represents the probability distribution of noise variance σ^2. Let $P_{y,x|\beta}$ denote the joint distribution of $x(t)$ and $y(t)$ for a given task β. Furthermore, for an unknown current task β, the input context to the transformer is given by a sequence of T_p independent and identically distributed (i.i.d.) pilots [12]

$$\mathcal{D}_{T_\mathrm{p}}^{\beta} = \big\{\big(y_\mathrm{p}(t), x_\mathrm{p}(t)\big)\big\}_{t=1}^{T_\mathrm{p}} \sim P_{y,x|\beta}^{\otimes T_\mathrm{p}}. \tag{9}$$

Due to the time-selective nature of the channel, the input prompt for a transformer becomes a function of time $t = T_\mathrm{p}+1, \ldots, T$, to capture the time-correlation between the channel at the tth time instant and the pilot context. We define the time-varying input prompt as

$$\begin{aligned} C_{\mathrm{BAS}}^{\beta}(t) &= \big(\mathcal{D}_{T_\mathrm{p}}^{\beta}, y_\mathrm{d}(t)\big) \\ &= \big(\big(y_\mathrm{p}(0), x_\mathrm{p}(0)\big), \ldots, \big(y_\mathrm{p}(T_\mathrm{p}), x_\mathrm{p}(T_\mathrm{p})\big), y_\mathrm{d}(t)\big). \end{aligned} \tag{10}$$

Furthermore, the transformer uses prompt (10) to produce a hypothesis $\widehat{x}(t)$ of the transmitted symbol $x(t)$ using the I/O relation defined as

$$\widehat{x}(t) = W_\Theta(C_{\mathrm{BAS}}^{\beta}(t)), \tag{11}$$

where W_Θ is the parameterized transformer mapping and Θ is the trainable model parameter. Unlike conventional methods (Sec. II-B), BASICSETS based symbol detection lacks explicit steps but aligns with the Bayesian prediction [9], [10] process, i.e., sampling $\widehat{x}(t)$ from the pre-trained distribution $P_{y,x|\beta}$ conditioned on the input prompt.

2) Symbol detection: Figure 1 illustrates the BASICSETS method for symbol detection, where the decoder-only transformer processes the input prompt (10) and generates a data symbol for the tth time instant. Here, multiple steps are involved with each step being parameterized by several trainable parameters, which collectively constitute the complete model parameter Θ.

Linear Input Embedding: Similar to a classification task in DL, symbol detection also involves selecting the most probable symbol from a discrete M-ary constellation set, $\mathcal{X}$. Thus, we represent $x(t)$ using a one-hot (OH) encoded vector [22], $\mathbf{x}^{\mathrm{oh}}(t) \in \mathbb{R}^{M\times 1}$, to feed the transformer. The OH transformation is represented using a bijective function, $F_{\mathrm{oh}}(.)$, as

$$\mathbf{x}^{\mathrm{oh}}(t) = F_{\mathrm{oh}}(x(t)). \tag{12}$$

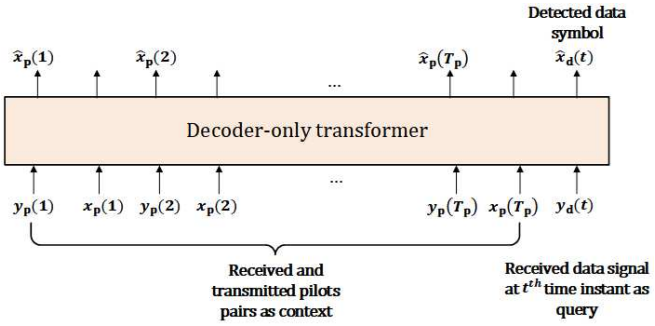

Fig. 1: Illustration of BASICSETS method for symbol detection.

Furthermore, we concatenate the real and imaginary parts of $y(t)$ to form $\tilde{\mathbf{y}}(t) = [\Re(y(t)), \Im(y(t))] \in \mathbb{R}^{2\times 1}$. To maintain uniform dimension across inputs in a prompt, a zero-padding of dimension $D_\mathrm{z} = \max(2, M)$ is applied to $\mathbf{x}^{\mathrm{oh}}(t)$ and $\tilde{\mathbf{y}}(t)$.

Let D_e be the dimension of embedding vector and $\mathbf{E}^0 \in \mathbb{R}^{D_\mathrm{e}\times(T_\mathrm{p}+1)}$ be the post-embedding sequence. The linear transformation performed by embedding layer is given by

$$\mathbf{E}^0 = \mathbf{W}_\mathrm{e}\cdot[\tilde{\mathbf{y}}_\mathrm{p}(1), \mathbf{x}_\mathrm{p}^{\mathrm{oh}}(1), \ldots, \tilde{\mathbf{y}}_\mathrm{p}(T_\mathrm{p}), \mathbf{x}_\mathrm{p}^{\mathrm{oh}}(T_\mathrm{p}), \tilde{\mathbf{y}}_\mathrm{d}(t)], \tag{13}$$

where $\mathbf{W}_\mathrm{e} \in \mathbb{R}^{D_\mathrm{e}\times D_\mathrm{z}}$ is a trainable embedding matrix.

Masked Multi-Head Self-Attention: The sequence $\mathbf{E}^0$ is processed through L stacked decoder layers, where each layer applies H self-attention heads, residual connections, layer normalization (LN) [23], and a feed-forward network (FFN) to obtain a transformed token $\mathbf{E}^l$ for $l = 1, 2, \ldots, L$. Each head $h = 1, 2, \ldots, H$ applies a key matrix $\mathbf{W}_h^\mathrm{k}$, a query matrix $\mathbf{W}_h^\mathrm{q}$ both $\in \mathbb{R}^{D_\mathrm{k}\times D_\mathrm{e}}$, and a value matrix $\mathbf{W}_h^\mathrm{v} \in \mathbb{R}^{D_\mathrm{v}\times D_\mathrm{e}}$ to obtain self-attention output [6], [12]

$$\mathbf{B}_h^l = \mathbf{W}_h^\mathrm{v}\mathbf{E}^{l-1}\cdot \mathrm{softmax}\left(\frac{(\mathbf{W}_h^\mathrm{k}\mathbf{E}^{l-1})^\top(\mathbf{W}_h^\mathrm{q}\mathbf{E}^{l-1})}{\sqrt{D_\mathrm{k}}}\right), \tag{14}$$

where $\mathbf{B}_h^l \in \mathbb{R}^{D_\mathrm{v}\times(T_\mathrm{p}+1)}$ and $D_\mathrm{k} = D_\mathrm{v} = D_\mathrm{e}/H$. The outputs of all heads are concatenated and projected using the weight matrix $\mathbf{W}_\mathrm{o} \in \mathbb{R}^{HD_\mathrm{v}\times D_\mathrm{e}}$, to obtain a multi-head attention output $\mathbf{A}^l$ given by

$$\mathbf{A}^l = \mathbf{W}_\mathrm{o}^\top[\mathbf{B}_1^l, \mathbf{B}_2^l, \ldots, \mathbf{B}_H^l] \in \mathbb{R}^{D_\mathrm{e}\times(T_\mathrm{p}+1)}. \tag{15}$$

The final output $\mathbf{E}^l \in \mathbb{R}^{D_\mathrm{e}\times(T_\mathrm{p}+1)}$ is produced by passing $\mathbf{A}^l$ through a two-layer FFN that has D_f number of hidden neurons, residual connections and LN function $F_1(\cdot)$, as

$$\mathbf{E}^l = \mathbf{W}_2F_2(\mathbf{W}_1F_1(\mathbf{E}^{l-1} + \mathbf{A}^l)) + \mathbf{E}^{l-1} + \mathbf{A}^l, \tag{16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{D_\mathrm{f}\times D_\mathrm{e}}$ and $\mathbf{W}_2 \in \mathbb{R}^{D_\mathrm{e}\times D_\mathrm{f}}$ are trainable weights of FFN, and $F_2(\cdot)$ is Gaussian error linear unit (GeLU) activation function [24].

Symbol Detection: The final layer output, $\mathbf{E}^L$, is passed through a linear layer with trainable weights $\mathbf{W}_\mathrm{s} \in \mathbb{R}^{M\times D_\mathrm{e}}$, to generate logits

$$\widetilde{\mathbf{x}}(t) = \mathbf{W}_\mathrm{s}\mathbf{E}^L \in \mathbb{R}^{M\times(T_\mathrm{p}+1)}. \tag{17}$$

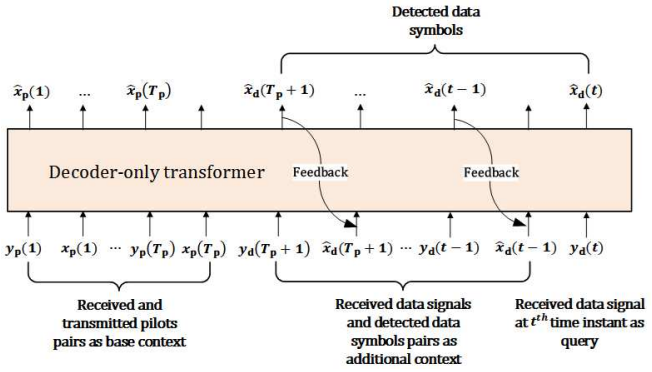

Fig. 2: Illustration of SEMINEXTSETS method for symbol detection.

A softmax classifier [22] is applied to the last token $\widetilde{\mathbf{x}}_{\rm d}(t)$ of $\widetilde{\mathbf{x}}(t)$, resulting in the OH encoded data symbol

$$\widehat{\mathbf{x}}_{\rm d}^{\rm oh}(t) = \arg\max_{\mathcal{X}}(\text{softmax}\big(\widetilde{\mathbf{x}}_{\rm d}(t)\big)) \in \mathbb{R}^{M\times 1}. \tag{18}$$

Using the inverse transform of (12), we obtain the complex detected symbol

$$\widehat{x}_{\rm d}(t) = F_{\rm oh}^{-1}(\widehat{\mathbf{x}}_{\rm d}^{\rm oh}(t)),\ \forall t = T_{\rm p}+1,\ldots,T. \tag{19}$$

Loss Function: The model parameter Θ is optimized using categorical cross-entropy (CCE) loss function $L_{\rm CCE}(.\,,.)$ [22]. For a batch size K, the total loss function is

$$L_{\rm BAS}(\Theta) = \frac{1}{MT}\sum_{m=1}^{K}\sum_{t=1}^{T} L_{\rm CCE}(\widehat{x}^m(t), x^m(t)), \tag{20}$$

where $\widehat{x}^m(t)$ and $x^m(t)$ denote detected and transmitted symbols for the mth sample in a batch and tth time instant, respectively. The optimal model parameter Θ^* is obtained by minimizing this loss, i.e.,

$$\Theta^* = \arg\min_{\Theta} L_{\rm BAS}(\Theta). \tag{21}$$

During model training, a gradient descent optimization technique [22] is used to iteratively solve (21).

B. Symbol detection using SEMINEXTSETS

The BASICSETS method suffers from low spectral efficiency (η) due to its high pilot-to-data ratio in time-selective channels. In addition, longer prompts exacerbate this issue, as the time-varying nature of the channel causes the fading coefficients to decorrelate between pilot and data symbols, undermining the detection accuracy. To address these limitations, we propose SEMINEXTSETS, a semi-blind ICL method that significantly reduces pilot overhead by leveraging detected symbols as an additional context [15]. This approach improves spectral efficiency and detection accuracy at the cost of increased computational complexity and inference latency.

1) Problem formulation: As shown in Fig. 2, SEMINEXTSETS augments pilot context with detected data symbols to produce a new time-varying prompt given by

$$\begin{aligned} C_{\rm SEM}^{\beta}(t) = \big(\mathcal{D}_{T_{\rm p}}^{\beta}, &\big(y_{\rm d}(T_{\rm p}+1), \widehat{x}_{\rm d}(T_{\rm p}+1)\big), \ldots, \\ &\big(y_{\rm d}(t-1), \widehat{x}_{\rm d}(t-1)\big), y_{\rm d}(t)\big), \end{aligned} \tag{22}$$

for $t = T_{\rm p}+1,\ldots,T$. Using (22), the transformer I/O relationship is written as

$$\widehat{x}(t) = W_{\Omega}(C_{\rm SEM}^{\beta}(t)), \tag{23}$$

where W_{Ω} represents the parameterized transformer mapping and Ω is the trainable model parameter. Furthermore, SEMINEXTSETS aligns with the semi-supervised learning (SSL) paradigm in DL [22]. Similar to SSL's challenges, this method contends with scarce labeled pilot context, necessitating the use of detected data symbols as noisy pseudo-labels under limited pilot availability. To maintain consistency with SSL principles, the CCE loss function excludes pilot symbols and focuses exclusively on data symbols, yielding the following comprehensive loss expression

$$L_{\rm SEM}(\Omega) = \frac{1}{MT_{\rm d}}\sum_{m=1}^{K}\sum_{t=T_{\rm p}+1}^{T} L_{\rm CCE}(\widehat{x}_{\rm d}^m(t), x_{\rm d}^m(t)). \tag{24}$$

Similar to (21), the optimal model parameter Ω^* for SEMINEXTSETS is obtained from

$$\Omega^* = \arg\min_{\Omega} L_{\rm SEM}(\Omega). \tag{25}$$

The gradient descent technique is used to iteratively solve the optimization problem (25).

2) Model training and symbol detection: For symbol detection at time t, SEMINEXTSETS requires $t-T_{\rm p}$ sequential forward passes, which are divided into two stages:

1) **Model-Freeze Stage**: For $t-T_{\rm p}-1$ passes, the model weights are frozen to exclude gradients from forward passes. This stage iteratively generates context using detected symbols, culminating in the final prompt (22).
2) **Model-Update Stage**: In the final pass, the model switches to training mode, enabling gradient computation for backpropagation to detect the tth symbol.

However, training the SEMINEXTSETS method from scratch incurs high computational costs due to $t-T_{\rm p}-1$ forward passes before the tth data symbol detection, which impacts both training and inference efficiency compared to BASICSETS. To mitigate this, we adopt a transfer learning approach:

1) **Pre-training**: Initialize the model to perform BASICSETS symbol detection until convergence of $L_{\rm BAS}$ in (20).
2) **Fine-tuning**: Transition to SEMINEXTSETS symbol detection using a balanced loss function defined as

$$L_{\rm FIT} = \gamma L_{\rm SEM}(\Theta) + (1-\gamma)L_{\rm BAS}(\Theta), \tag{26}$$

where $\gamma \in [0,1]$ controls the emphasis on SEMINEXTSETS ($\gamma \uparrow 1$) versus BASICSETS ($\gamma \downarrow 0$).

C. Symbol detection using GENIESETS

SEMINEXTSETS relies on detected symbols for context, which introduces error propagation risks for subsequent symbol detection. To quantify this effect, we propose GENIESETS, a hypothetical benchmark that assumes perfect

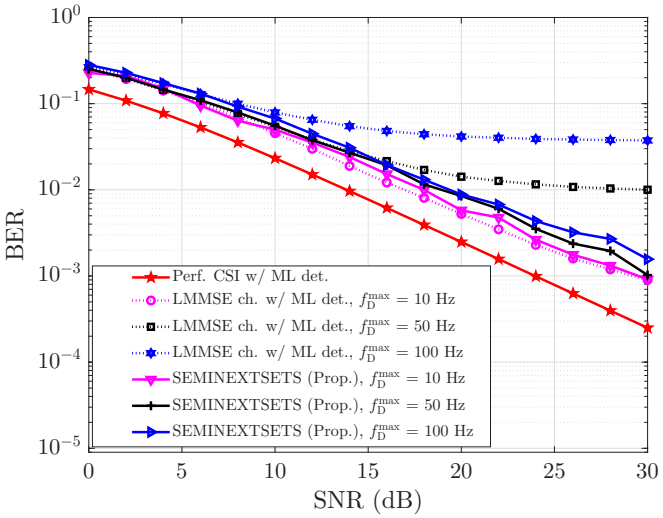

Fig. 3: BER vs SNR performance plots for BPSK modulation.

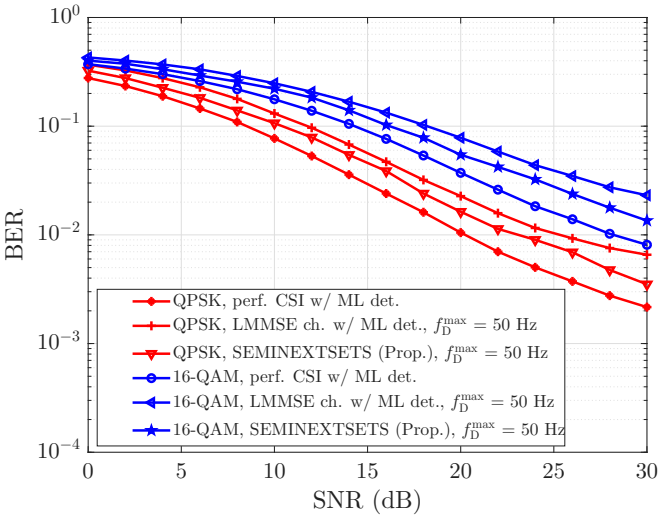

Fig. 4: BER vs SNR performance plots for QPSK and 16-QAM.

knowledge of transmitted data symbols. This method replaces detected symbols $\widehat{x}_{\mathrm{d}}(\cdot)$ with true symbols $x_{\mathrm{d}}(\cdot)$ in context (22), eliminating error propagation. The updated prompt structure is defined as

$$C_{\mathrm{GEN}}^{\beta}(t) = \big(\mathcal{D}_{T_{\mathrm{p}}}^{\beta}, \big(y_{\mathrm{d}}(T_{\mathrm{p}}+1), x_{\mathrm{d}}(T_{\mathrm{p}}+1)\big), \ldots, \big(y_{\mathrm{d}}(t-1), x_{\mathrm{d}}(t-1)\big), y_{\mathrm{d}}(t)\big). \tag{27}$$

As will be shown later, GENIESETS lower-bounds the BER performance of SEMINEXTSETS by isolating the impact of error propagation.

IV. Results and Discussions

In this section, we analyze the performance of SEMINEXTSETS against conventional symbol detection methods and GENIESETS for different modulation schemes over SISO time-selective channels with different Doppler shifts. We employ $T = 31$ prompt length, $T_{\mathrm{p}} = 1$ pilot, and $T_{\mathrm{d}} = 30$ data symbols for both training and inference. The SNR values are uniformly sampled from $[0, 30]$ dB, while maximum Doppler shifts $f_{\mathrm{D}}^{\max}$ are drawn from $[0, 100]$ Hz, ensuring that the model encounters diverse noise and fading conditions during pre-training and fine-tuning. The time-selective channel fading coefficients are generated using Clarke's model [17].

The transformer configuration uses $D_{\mathrm{e}} = 64$ embedding dimensions, $L = 8$ layers, and $H = 8$ self-attention heads per layer. We employ Adam optimization [25] with a learning rate of 10^{-4}, batch size of 512, and 8 training epochs. For fine-tuning SEMINEXTSETS, we set $\gamma = 0.8$ in (26) to prioritize SEMINEXTSETS performance while retaining stability of BASICSETS. The dataset comprises 7.68 million training samples, 1,000 validation samples, and 250,000 test samples, ensuring statistical robustness in performance evaluation.

In Fig. 3, perfect-CSI with ML detection (Sec. II-B1) serves as an optimal BER benchmark. SEMINEXTSETS achieves performance comparable to that of LMMSE channel estimation with ML detection (Sec. II-B2) at $f_{\mathrm{D}}^{\max} = 10$ Hz and

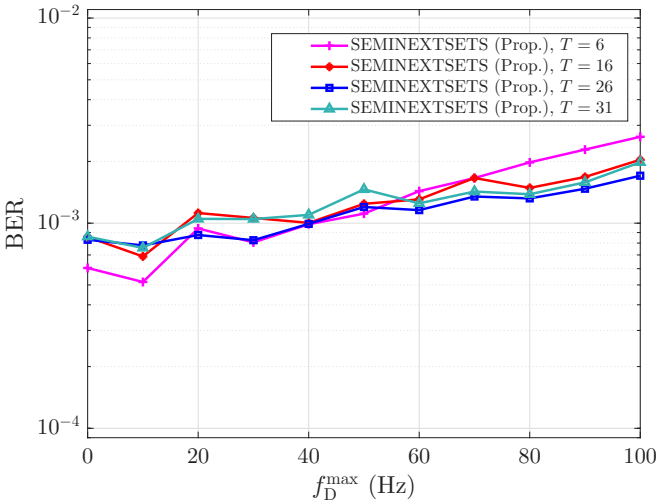

Fig. 5: BER vs maximum Doppler shift plots for SEMINEXTSETS method with different prompt lengths T for BPSK at SNR = 30 dB.

significantly outperforms it at $f_{\mathrm{D}}^{\max} = 50$ Hz and 100 Hz. Notably, SEMINEXTSETS exhibits minimal BER variation across these Doppler shifts, demonstrating its robustness in tracking and equalizing channel variations across diverse SNR and Doppler conditions, justifying the increase in computational overhead.

Figure 4 demonstrates that SEMINEXTSETS achieves competitive BER performance for both QPSK and 16-QAM modulations across low and high SNRs. Consistent with BPSK results, it outperforms LMMSE channel estimation with ML detection under challenging time-selective fading, while preserving spectral efficiency through minimal pilot overhead. These results validate the approach's practical viability for deployment in dynamic wireless environments where modulation schemes may need to be dynamically selected based on channel conditions.

Figure 5 demonstrates that BER increases with higher Doppler shifts ($f_{\mathrm{D}}^{\max}$), consistent with theoretical expectations. Notably, we do not observe any systematic performance ad-

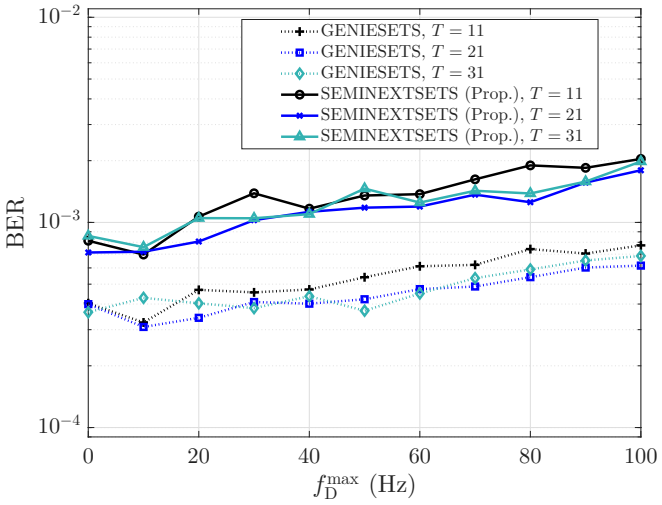

Fig. 6: BER performance comparison between SEMINEXTSETS and GENIESETS methods for BPSK and SNR = 30 dB.

vantage emerging between prompt lengths ($T = 6, 16, 26, 31$), as marginal BER variations favor shorter or longer prompts depending on specific Doppler conditions. This indicates that the symbol detection accuracy of SEMINEXTSETS is not compromised with increasing prompt length. However, channel usage efficiency η scales with prompt length: 0.833 ($T = 6$), 0.9375 ($T = 16$), 0.962 ($T = 26$), and 0.968 ($T = 31$). These findings confirm that longer prompts with single-pilot configurations enhance spectral efficiency while maintaining competitive BER performance across diverse fading conditions.

Finally, Fig. 6 shows that GENIESETS significantly outperforms SEMINEXTSETS, with the performance gap primarily attributed to error propagation in SEMINEXTSETS. This establishes GENIESETS as a lower-bound benchmark for SEMINEXTSETS BER. Furthermore, fluctuations in both BER curves may result from the transformer's reliance on ICL to infer patterns from prompts, rather than explicit training or fine-tuning on unknown data distributions.

V. Conclusions

We investigated transformer-based in-context learning for symbol detection in time-selective fading channels, a previously unexplored area. Our proposed SEMINEXTSETS method for symbol detection adopts a transformer-based semi-blind ICL approach, which uses a time-varying augmented prompt derived from limited pilots and detected symbols, enhancing detection under dynamic channel conditions. Through transfer learning, pre-training with BASICSETS and fine-tuning using a balanced loss function, we achieved efficient model convergence. Simulations demonstrated competitive BER performance for BPSK, QPSK, and 16-QAM modulations across diverse SNRs (0-30 dB) and Doppler shifts (0-100 Hz), outperforming conventional LMMSE-based methods at higher Doppler frequencies and SNR conditions. Spectral efficiency (η) reached 0.968 with 30 data symbols per frame, showing negligible BER degradation compared to shorter prompts. Using GENIESETS as a lower-bound benchmark, we quantified error propagation effects and identified BER fluctuations attributable to ICL's contextual learning. Future work can focus on investigating BER stability and error propagation mitigation, alongside exploring model compression to reduce computational overhead.

References

[1] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Letters*, vol. 23, no. 4, pp. 652–655, Feb. 2019.

[2] S. R. Mattu, L. N. Theagarajan, and A. Chockalingam, "Deep channel prediction: A DNN framework for receiver design in time-varying fading channels," *IEEE Trans. Veh. Tech.*, vol. 71, no. 6, pp. 6439–6453, Mar. 2022.

[3] M. Honkala, D. Korpi, and J. M. J. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3925–3940, Feb. 2021.

[4] F. Ait Aoudia and J. Hoydis, "End-to-end learning for ofdm: From neural receivers to pilotless communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1049–1063, Aug. 2022.

[5] N. Shlezinger and Y. C. Eldar, "Model-based deep learning," *arXiv:2306.04469*, Jun. 2023.

[6] A. Vaswani *et al.*, "Attention is all you need," *Proc. NIPS 2017*, vol. 30, pp. 6000–6010, Dec. 2017.

[7] T. Brown *et al.*, "Language models are few-shot learners," *Proc. NIPS 2020*, vol. 33, pp. 1877–1901, Dec. 2020.

[8] S. Garg, D. Tsipras, P. Liang, and G. Valiant, "What can transformers learn in-context? a case study of simple function classes," *Proc. NIPS 2022*, vol. 35, pp. 30 583–30 598, Nov. 2022.

[9] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An explanation of in-context learning as implicit bayesian inference," *Proc. ICLR 2022*, Jan. 2022.

[10] M. Panwar, K. Ahuja, and N. Goyal, "In-context learning through the bayesian prism," *Proc. ICLR 2024*, Jan. 2024.

[11] V. T. Kunde *et al.*, "Transformers are provably optimal in-context estimators for wireless communications," *Proc. AISTATS 2025*, vol. 258, pp. 1531–1539, Jan 2025.

[12] M. Zecchin, K. Yu, and O. Simeone, "In-context learning for mimo equalization using transformer-based sequence models," *IEEE ICC Workshops*, pp. 1573–1578, Jun. 2024.

[13] Z. Song, M. Zecchin, B. Rajendran, and O. Simeone, "Turbo-icl: In-context learning-based turbo equalization," *arXiv:2505.06175*, May 2025.

[14] M. Abbas, K. Kar, and T. Chen, "Leveraging large language models for wireless symbol detection via in-context learning," *Proc. IEEE GLOBECOM 2024*, pp. 5217–5222, Dec. 2024.

[15] L. Fan, J. Yang, C. Shen, and C. L. Brown, "Decision feedback in-context symbol detection over block-fading channels," *Proc. IEEE ICC 2025*, pp. 3785–3790, Jun. 2025.

[16] R. H. Clarke, "A statistical theory of mobile-radio reception," *The Bell System Technical Journal*, vol. 47, no. 6, pp. 957–1000, Jul. 1968.

[17] J. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Trans. Veh. Tech.*, vol. 24, no. 3, pp. 39–40, Aug. 1975.

[18] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. Cambridge University Press, 2014.

[19] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., 1993.

[20] H. Arslan and G. E. Bottomley, "Channel estimation in narrowband wireless communication systems," *Wireless Commun. and Mob. Comput.*, vol. 1, no. 2, pp. 201–219, Mar. 2001.

[21] A. Radford *et al.*, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019. [Online]. Available: https://openai.com/research/better-language-models

[22] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. MIT press, 2022.

[23] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, Jul. 2016.

[24] D. Hendrycks, "Gaussian error linear units (GeLUs)," *arXiv:1606.08415*, Jun. 2016.

[25] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, Dec. 2014.