Computer Networks and Communications
http://ojs.wiserpub.com/index.php/CNC/

Research Article

A Deep-SIC Channel Estimator Scheme in NOMA Network

Sumita Majhi, Kaushal Shelke, Pinaki Mitra*

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India
* Correspondence: pinaki@iitg.ac.in

Received: 21 April 2026; **Revised:** 26 June 2026; **Accepted:** 2 July 2026; **Published:** 24 August 2026

Abstract: In 5G and next-generation mobile networks, reliable handover is a key requirement that ensures connection continuity, especially for mobile users and in high-density scenarios. However, conventional handover triggers based on instantaneous channel measurements are prone to failures and the ping-pong effect due to outdated or inaccurate channel state information. To address this, we propose Deep Successive Interference Cancellation (Deep-SIC), a knowledge-based channel prediction model that employs a Transformer-based approach to estimate channel quality and optimise handover decisions. Deep-SIC utilises Partially Decoded Data (PDD), a byproduct of SIC in Non-Orthogonal Multiple Access (NOMA), as a feedback signal to continually improve its predictions. This mechanism enables the model to learn quickly and stabilise their learning. Our model learns faster than existing state-of-the-art algorithms under the evaluated conditions, such as Graph-NOMA, while providing verifiable stability and resilience to user mobility (Theorem 2). The system-level simulation can substantially enhance network performance by mitigating the ping-pong effect and reducing the handover failure (HOF) rate, especially at vehicular speeds (e.g., 60 km/h). Moreover, Deep-SIC has a smaller normalised root mean square error (NRMSE) in low-SNR situations than state-of-the-art algorithms with linear computational complexity, $O(K)$.

Keywords: Non-Orthogonal Multiple Access (NOMA), channel prediction, handover failure rate, ping pong rate, transformer

1. Introduction

The proliferation of mobile devices and the advent of IoT have made seamless connectivity a cornerstone of modern wireless systems. In cellular networks, user mobility necessitates frequent handovers between base stations (BSs). These handovers cause dropped connections, poor quality of service (QoS) and the ping-pong effect. The primary cause of these failures is reliance on outdated channel state information (CSI) to make handover decisions, which fail to account for the rapid temporal changes caused by user movement and fading. Conventional handover algorithms are triggered by simple thresholds on metrics such as Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ) or Signal-to-Interference-plus-Noise Ratio (SINR) from both serving and neighbouring BSs.

Recent studies have explored Machine Learning (ML) for channel estimation, mainly focusing on denoising or static scenarios. They are not aimed at the long-term forecasting which is an important factor to handover proactively. Moreover, the issue is worsened in NOMA networks, where the interference at the receiver is complicated, and channel prediction becomes even more difficult. To address these shortcomings, we present Deep-SIC, a novel framework for intelligent handover management. The main contributions of this work are summarized as follows:

Copyright ©2026 Pinaki Mitra, et al.
DOI: https://doi.org/10.37256/cnc.42202610053
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

• A predictive handover management framework (Deep-SIC) that uses a Transformer model for accurate short-term channel forecasting in mobile environments.

• A closed-loop feedback system with PDD is used to make the prediction more robust and minimise the effects of interference cancellation errors.

• Theoretical convergence and mobility resilience validate the model's stability under dynamic network conditions.

• System-level simulations demonstrate that Deep-SIC substantially reduces handover failure rates and ping-pong effects, while improving spectral efficiency, which is a key metric for network operators.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the motivation and contributions. Section 4 introduces the system model. Section 5 describes the Deep-SIC framework. Section 6 discusses results, Section 7 provides theoretical analysis, and Section 8 concludes the paper.

2. Related work

Classical estimators such as the Minimum Mean Square Error (MMSE) and Least Squares (LS) remain a reference in the literature [1, 2]. MMSE achieves optimality under Gaussian noise when the channel statistics are known, but its implementation often requires large matrix inversions, which become costly in high-dimensional MIMO settings. The LS estimator is computationally simpler and model-agnostic, yet it degrades remarkably in low-SNR regimes and is more susceptible to interference and noise. Both assume static or slowly varying channels and lack predictive capability for dynamic handover scenarios.

Pilot-based methods use known reference signals for channel estimation. Recent advances include Convolutional Neural Network-integrated approaches [3], LS-based estimators for active RIS [4], and tensor-decomposition methods [5]. These are effective in static conditions but suffer from high pilot overhead, sensitivity to modelling mismatches, and rapid accuracy degradation under fast user mobility, such as in vehicular networks. At vehicular speeds, the pilot transmission interval is longer than the channel coherence time, leading channel estimates to become outdated before handover decisions can be made.

Blind methods utilise signal statistics in the absence of pilots, employing subspace-based techniques [6–9] and higher-order statistics [10–12]. Although these approaches do not require pilots, their computational complexity $O(N^3)$ [12] renders them impractical for real-time handover. Furthermore, their slow convergence and high sensitivity to noise are especially challenging in the interference-rich environment, a key characteristic of the NOMA network.

Semi-blind methods achieve a balance between pilot overhead and data-driven information. Examples include antenna-partitioned schemes [13], reinforcement learning estimators [14], and iterative virtual-pilot schemes [15]. These methods lack mechanisms for assessing detection reliability, which renders them unsuitable for high-mobility scenarios characterised by frequent errors.

Deep learning architectures such as Denoising CNNs (DnCNNs) [16] demonstrate potential for channel denoising. Nevertheless, their localised receptive fields are insufficient to capture long-range temporal dependencies arising from user mobility. Moreover, since these models process frames independently, they disregard the channel's temporal evolution, which is essential for enabling proactive handover. Without temporal modelling, they cannot predict future channel conditions.

Research on NOMA handover (NOMA-HO) is currently limited. Existing studies [17–20] primarily address CSI estimation using conventional methods that are not intended for handover control. Practical challenges such as HOF and ping-pong effects persist [19, 21]. To date, no NOMA-HO algorithm incorporates temporal prediction or SIC feedback correction. Current channel estimation methods as demonstrated in Figure 1 exhibit fundamental limitations when applied to NOMA-HO in high-mobility environments. Based on existing channel estimation methods, four categories can be identified: pilot-based, blind, semi-blind, and deep learning methods, as shown in Figure 1. Pilot-based approaches, such as MMSE, LS, and CNN-integrated estimators, require reference signals, resulting in high overhead and poor performance under high mobility. In contrast, blind methods (subspace, higher-order statistics) require no pilots but are computationally intensive ($O(N^3)$ and converge slowly. Semi-blind schemes offer a compromise but do not provide a reliability assessment

for SIC errors. Deep learning architectures, for instance, DnCNN, operate independently on frames and cannot capture the channel's temporal evolution. Overall, while these approaches aim to predict future CSI, none leverage SIC feedback, as discussed further in the limitations below, thereby underscoring the need for the proposed Deep-SIC framework. Section 3 presents a comprehensive analysis of these constraints and demonstrates the necessity for the Deep-SIC framework.

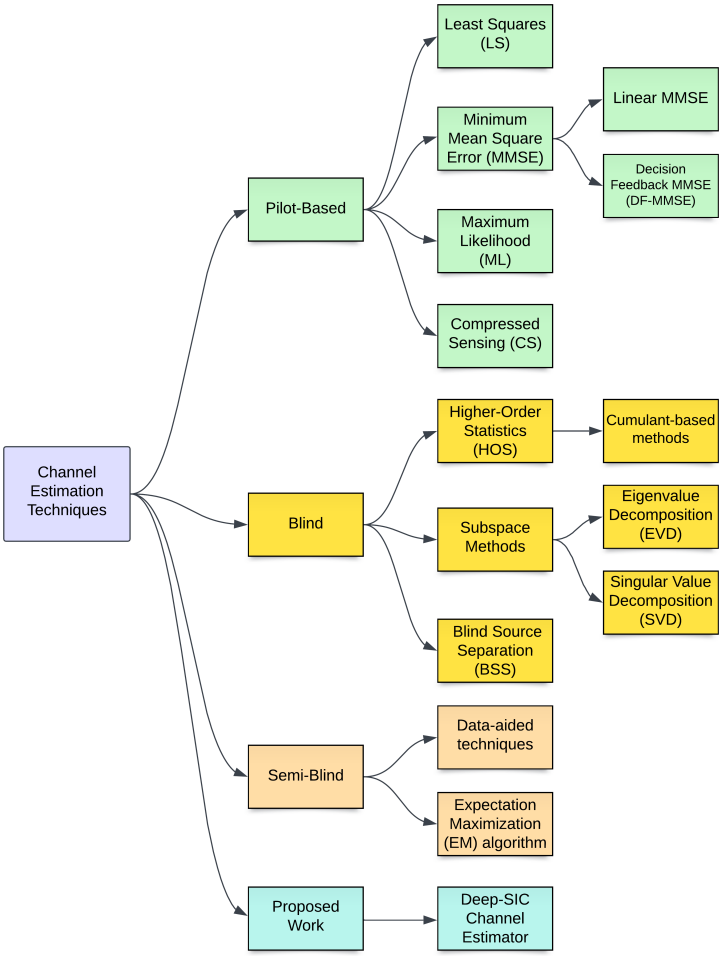


Figure 1. Classification of channel estimation techniques

3. Motivation and contributions

Handover failures (HOF) and the ping-pong effects are primary causes of dropped connections and poor user experience in mobile networks. While deep learning models like DnCNN [16] can mitigate static impairments, they are unable to model the long-term temporal evolution of the channel caused by user mobility, precisely the information needed to predict and prevent a faulty handover. Existing methods fail in high-mobility NOMA scenarios for five key reasons:

- They estimate current channels rather than predicting future conditions, making proactive handover impossible.
- They assume channel coherence times longer than handover intervals, invalid at vehicular speeds (60–120 km/h).
- CNNs lack mechanisms to learn temporal patterns, ignoring channel evolution.
- None of the existing methods leverage SIC byproducts (PDD) as feedback, treating residual interference as noise.
- They optimise for estimation accuracy rather than handover performance (HOF, ping-pong rate).

These gaps are addressed by combining a Transformer-based model for predicting future channel conditions with a closed-loop PDD feedback system for continuous prediction correction. In contrast to previous methods that do not utilise self-attention or closed-loop correction, the Transformer's self-attention mechanism captures long-range dependencies, while PDD enables real-time error correction. Consequently, this integrated approach supports proactive and reliable handover decisions in dynamic environments. To the best of our knowledge, no prior work [19, 21–25] has combined a Transformer with PDD-based feedback for channel prediction and handover reduction in NOMA networks. Our contributions are listed below.

• We propose Deep-SIC, a Transformer-based channel estimation model that captures long-range temporal dependencies to enable accurate short-term CSI forecasting in mobile environments.

• The key enabler is a novel closed-loop feedback system that leverages PDD from the SIC process to continuously correct and refine channel predictions.

• We provide theoretical validations of our model: convergence conditions (Proposition 1), estimation error bounds (Theorem 1), a mobility resilience bound (Theorem 2), and BER improvement (Lemma 1).

• We demonstrate that our approach is practically feasible, scaling with linear complexity $O(K)$ in the number of users (Proposition 2, Figure 11), and leverage transfer learning to ensure robust performance even with limited data.

• The proposed approach reduces handover failure rates as compared to the existing state of the art. It also significantly mitigates ping-pong effects and achieves a lower normalised root-mean-square error (NRMSE) in low-SNR conditions.

4. System model

We consider a downlink NOMA cellular network depicted in Figure 2, where User Equipments (UEs) must maintain connectivity while moving. A single BS equipped with M antennas serves K single-antenna UEs. The channel from the BS to the k-th UE is a complex vector $\mathbf{h}_k \in \mathbb{C}^{M\times 1}$, which we model as Rayleigh fading to capture the signal variations typical in mobile scenarios, i.e., $\mathbf{h}_k \sim \mathscr{CN}(\mathbf{0}, \mathbf{I}_M)$. The BS sends a superposed signal

$$\mathbf{x} = \sum_{k=1}^{K} \sqrt{P_k}\, \mathbf{w}_k s_k, \tag{1}$$

where s_k denotes the data symbol, P_k the transmit power, and $\mathbf{w}_k \in \mathbb{C}^{M\times 1}$ the unit-norm precoding vector for user k ($\|\mathbf{w}_k\|^2 = 1$). The received signal at user k is

$$y_k = \mathbf{h}_k^H \mathbf{x} + n_k, \tag{2}$$

or equivalently

$$y_k = \mathbf{h}_k^H \left(\sum_{i=1}^{K} \sqrt{P_i}\, \mathbf{w}_i s_i \right) + n_k, \tag{3}$$

with $n_k \sim \mathscr{CN}(0, \sigma_n^2)$ modeling additive white Gaussian noise (AWGN).

We order users by their effective channel gains:

$$\left|\mathbf{h}_1^H \mathbf{w}_1\right|^2 \geq \left|\mathbf{h}_2^H \mathbf{w}_2\right|^2 \geq \cdots \geq \left|\mathbf{h}_K^H \mathbf{w}_K\right|^2. \tag{4}$$

Following the NOMA protocol, the strongest user (User 1) performs SIC. User 1 first decodes the signals intended for weaker users (Users 2 through K), subtracts those estimates from its received signal, and then decodes its own symbol. A weaker user $k > 1$ treats signals from stronger users as interference and decodes its symbol directly.

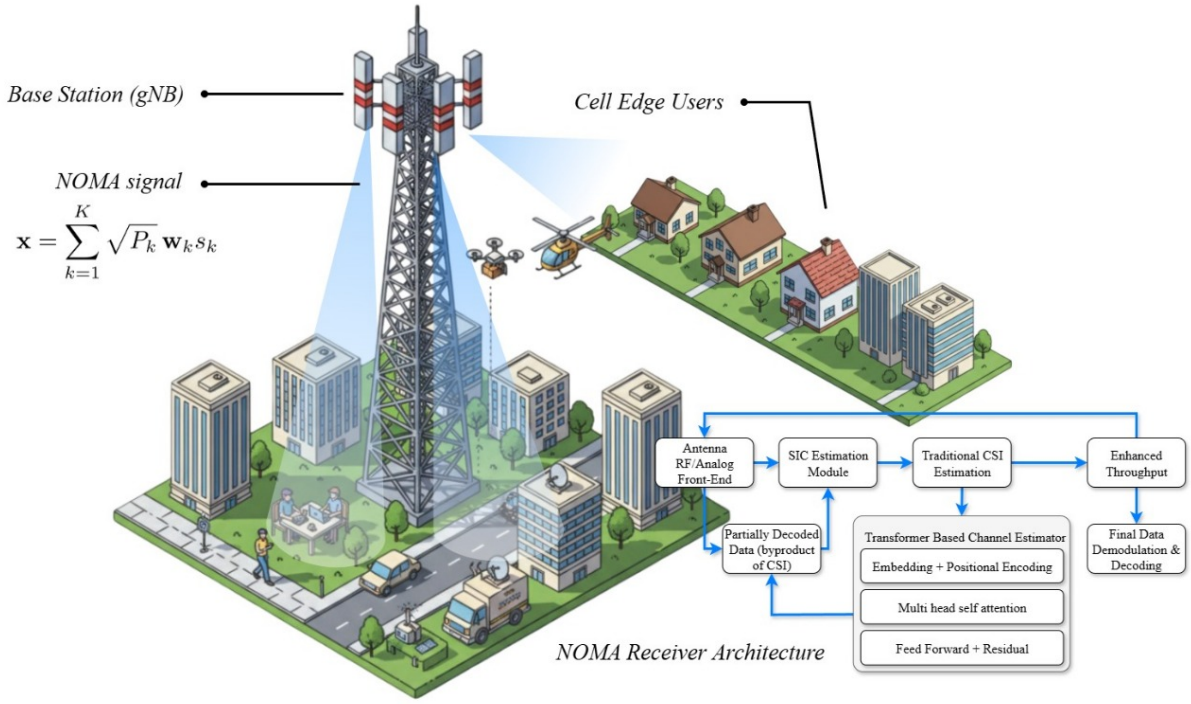


Figure 2. System model and receiver architecture

4.1 *Imperfect SIC and PDD*

SIC relies on good channel estimates $\hat{\mathbf{h}}_k$. When these estimates are imperfect, cancellation leaves residual interference. To capture this effect, we introduce PDD, a byproduct of CSI in NOMA channel estimation. Let $\tilde{s}_i^{(k)}$ be the symbol for user i as decoded by user k during SIC. For example, when User 1 attempts to decode User 2, the decoding error is

$$\widetilde{s_2} = s_2 - \tilde{s}_2^{(1)}, \tag{5}$$

so $\widetilde{s_2}$ denotes the residual, undecoded component (PDD). This residual interference not only degrades the user's immediate signal decoding but also corrupts the channel quality estimates used for long-term network management tasks like handover. Therefore, we propose to use this PDD as a feedback signal to correct the initial channel estimate, thereby improving the overall stability of the network's view of the channel. After attempting to cancel User 2's contribution, the received signal at User 1 becomes

$$y_{1,\mathrm{SIC}} = \mathbf{h}_1^H \left(\sqrt{P_1}\, \mathbf{w}_1 s_1 + \sqrt{P_2}\, \mathbf{w}_2 \widetilde{s_2} \right) + \sum_{i=3}^{K} \mathbf{h}_1^H \sqrt{P_i}\, \mathbf{w}_i s_i + n_1. \tag{6}$$

The term $\sqrt{P_2}\,\mathbf{w}_2\tilde{s}_2$ is the residual interference that degrades User 1's decoding. Rather than treating this term as pure noise, we view $\tilde{s}_2$ as a useful signal that carries information about the error in the channel estimate. We use this PDD as an additional feedback measure to improve the initial channel estimate.

For a weaker user (e.g., User 2), the received signal is

$$y_2 = \mathbf{h}_2^H \sqrt{P_1}\,\mathbf{w}_1 s_1 + \mathbf{h}_2^H \sqrt{P_2}\,\mathbf{w}_2 s_2 + \sum_{i=3}^{K} \mathbf{h}_2^H \sqrt{P_i}\,\mathbf{w}_i s_i + n_2, \tag{7}$$

and User 2 must treat the first term as interference since it does not perform SIC.

4.2 *NOMA Handover (NOMA-HO) Scenario*

Robust channel estimates are especially important during handover. We consider a two-cell setup where a far user U_0^F currently served by BS_0 may hand over to a neighboring cell served by BS_1. A handover decision typically compares channel-quality metrics such as RSRQ derived from CSI. We improve this choice by considering the stability of each estimate using PDD. The efficiency of this system is assessed in a mobility situation where a user is required to decide to pass on the serving BS to a target BS, where the accuracy and predictability of the channel estimates h_k hold the success of the process. In practice, we cause a handover when the adjusted quality of the target BS is better than the quality of the serving BS:

$$\mathrm{CSI}_1 + \alpha \cdot \mathrm{PDD}_1 \;>\; \mathrm{CSI}_0 + \alpha \cdot \mathrm{PDD}_0, \tag{8}$$

where CSI_j represents the estimated quality of all the channels of BS j, PDD_j represents the measurement of estimation reliability (a smaller value signifies a more reliable estimate), and α is a scaling factor to reduce compromises between quality and reliability. Under this rule, the network assigns users to cells with stronger signals, as well as to cells whose channel estimations are more reliable, thereby reducing ping-pong effects and failures during handover.

5. Proposed work

The Deep-SIC framework is an intelligent processing module in the network stack that supports handover decisions. It operates in two primary stages: initially, it performs fundamental channel estimation using regular pilots. Then it enters a refinement loop, where it utilises the data transmission itself to refine the estimation. It is this self-correcting mechanism that creates a closed loop, allowing the network to maintain a reliable and predictive view of the channel state, thereby enabling timely handovers.

Figure 3 shows the overall flow of the Deep-SIC framework, implemented in the following sequential steps: initial channel estimation, SIC decoding, PDD feedback, Transformer optimisation, and handover decision-making. The core estimation pipeline (Algorithm 1) is separated from the transfer learning adaptation (Algorithm 2), and the data flow and feedback loop between the modules are explicitly depicted in the flowchart.

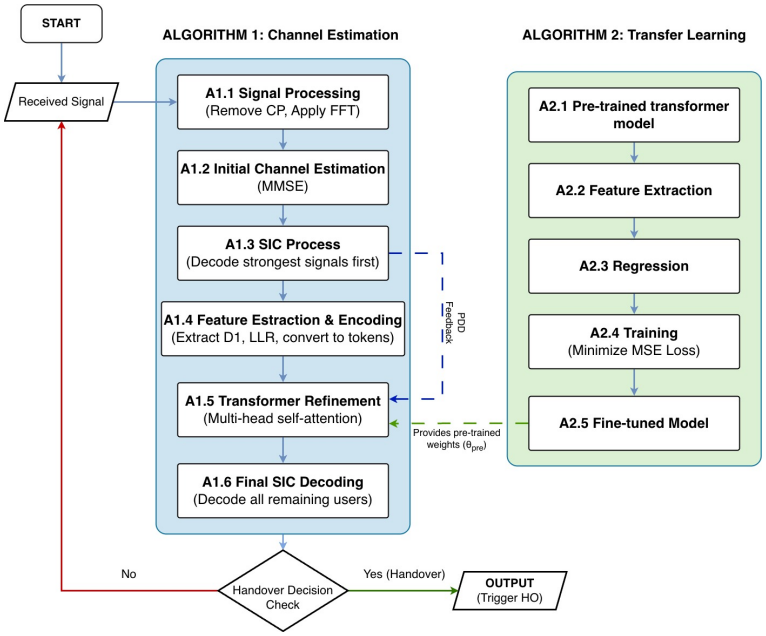


Figure 3. Clarify division between Algorithm 1 and Algorithm 2, show data flow and feedback loops

5.1 *Deep-SIC channel estimator model*

The proposed algorithm (Algorithm 1) is a data-aided algorithm that utilises deep learning to effectively and efficiently estimate channels in NOMA networks. It utilises PDD symbols in the estimation, leveraging better available information to achieve greater accuracy. The Transformer network is the most crucial element that enables prediction capabilities. The accuracy, however, is not limited to its self-attention mechanism alone, but considers temporal forecasting. It can predict the quality of the future channel by taking into account the significance of past channel measurements, and the handover algorithm can operate proactively instead of reactively. It is the essence of its superiority in decreasing HOF and ping-pong rates.

Algorithm 1 describes the process of channel estimation using a Transformer-based data-aided data estimation approach. It begins with signal reception, followed by an initial estimate of the channel using pilot-based algorithms such as MMSE. The estimate of the MMSE for the k-th user is as follows.

$$h_{k,\mathrm{MMSE}} = (R_{hh} + nR_{ss}^{-1})^{-1} R_{hh} s_k^H y_k, \tag{9}$$

In which the channel covariance matrix is denoted as R_{hh}, the pilot symbol autocorrelation matrix is denoted as R_{ss}, n is the noise variance, and s_k^H is the Hermitian transpose of the pilot vector.

SIC is then used to decode the strongest user's symbol d_1. Its log-likelihood ratio is then calculated as

$$LLR_1 = \log\left(\frac{P(d_1 = 1 \mid y(t), h_1)}{P(d_1 = 0 \mid y(t), h_1)}\right). \tag{10}$$

In contrast to the sequential models based on RNN, this approach transforms temporal features (decoded symbols, LLRs, and initial CSI) into positional-encoded token sequences. This eventually maintains temporal relationships and provides parallel processing.

The tokens are then passed through multi-head self-attention layers:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{11}$$

Q, K and V are query, key and value matrices obtained from the tokens. This process encapsulates long-range dependencies and contextual relationships throughout the sequences. The Transformer refines the estimates through deep learning, producing improved values $\hat{h}_k^{\text{transformer}}$ for all users.

Finally, these refined estimates are used in the next stages of SIC to decode the signals of the remaining users. The algorithm outputs decoded data $d_1, d_2, \ldots, d_K$, demonstrating improved performance in modeling channel dynamics and interference.

Algorithm 1 Transformer-Based Data-Aided Channel Estimation in NOMA

Input: Received signal $y'(t)$, pilot symbols $s_k(t)$ for $k \in \{1,\ldots,K\}$
Output: Decoded data $d_1, d_2, \ldots, d_K$
Receive Signal Processing: Remove the cyclic prefix (CP) and apply FFT.
Initial Channel Estimation: Use MMSE to obtain initial estimates:

$$h_{k,\text{MMSE}} = (R_{hh} + nR_{ss}^{-1})^{-1} R_{hh} s_k^H y_k.$$

SIC Process: Apply SIC iteratively to decode the strongest signals first.
Feature Extraction: Decode the strongest user's symbol d_1 and compute its LLR:

$$LLR_1 = \log\left(\frac{P(d_1 = 1 \mid y(t), h_1)}{P(d_1 = 0 \mid y(t), h_1)}\right).$$

Feature Encoding: Convert decoded symbols, LLRs, and initial CSI into token sequences with positional encoding.

Transformer Refinement: Pass tokens through multi-head self-attention layers:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

Channel Estimation: Use feed-forward layers to obtain refined estimates $\hat{h}_k^{\text{transformer}}$.
Decoding: Apply refined estimates in subsequent SIC stages to decode all remaining user signals.

5.2 *Data preparation*

To ensure our model is practical and can be deployed in scenarios with limited local data, we employ a transfer learning strategy (Algorithm 2). This allows a pre-trained Transformer model, which has learned general temporal patterns of wireless channels, to be quickly adapted with minimal data to a new network environment. This reduces the operational overhead of collecting massive training datasets for every cell, making the system highly adaptable. Figure 4 shows the workflow. Data augmentation (Figure 5, source: [26]) further enhances generalization, using a KDE plot to highlight the smoothing effect on the distribution. The augmented dataset will be released in [27]. Computational complexity is analyzed in Section 7.5, Proposition 2.

Algorithm 2 Transfer Learning for CSI Prediction with Transformer

Input: Pre-trained Transformer model $\mathbf{M}_{\text{pre}}(\mathbf{X};\boldsymbol{\theta}_{\text{pre}})$; training set $\{\mathbf{X}_i,\mathbf{y}_i\}$ with $\mathbf{X}_i=[\text{RSRQ},\text{CQI},\text{PDD},\text{SNR}]$, $\mathbf{y}_i=[\text{RSRQ},\text{SNR}]$

Output: Fine-tuned regression model $\mathbf{M}_{\text{new}}(\mathbf{h};\boldsymbol{\theta}_{\text{new}})$

FEATURE EXTRACTION: Pass $\mathbf{X}$ through the pre-trained Transformer to obtain embeddings:

$$\mathbf{H}=\mathbf{M}_{\text{pre}}(\mathbf{X};\boldsymbol{\theta}_{\text{pre}}).$$

REGRESSION: Apply a regressor on pooled embeddings $\mathbf{H}$ (e.g., mean pooling or [CLS] token) to predict $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}}=\mathbf{M}_{\text{new}}(\text{Pool}(\mathbf{H});\boldsymbol{\theta}_{\text{new}}).$$

TRAINING: Minimize the MSE loss:

$$\mathscr{L}(\hat{\mathbf{y}},\mathbf{y})=\tfrac{1}{2}\left[(\text{RSRQ}-\widehat{\text{RSRQ}})^2+(\text{SNR}-\widehat{\text{SNR}})^2\right].$$

Note: The pre-trained Transformer weights are frozen; only $\boldsymbol{\theta}_{\text{new}}$ is updated.

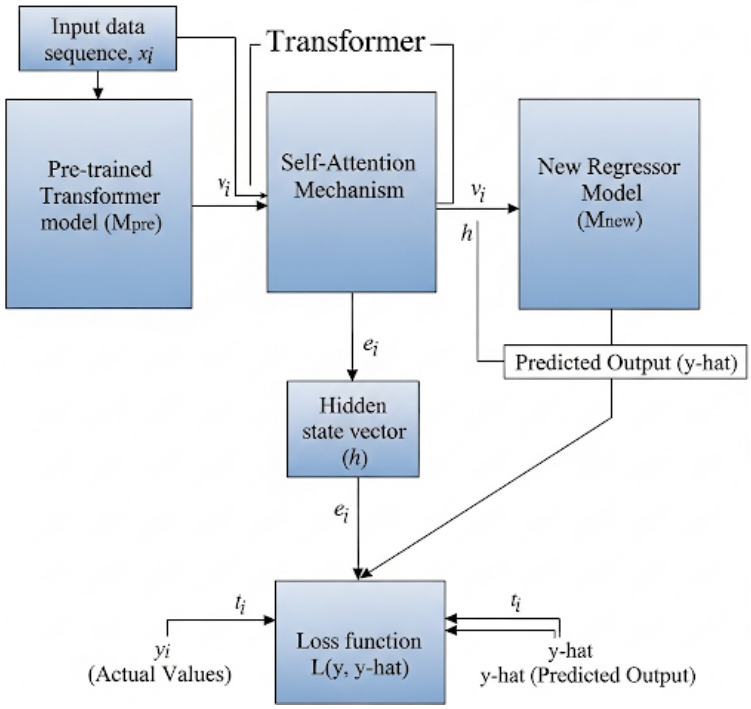


Figure 4. Flowchart of Algorithm 2

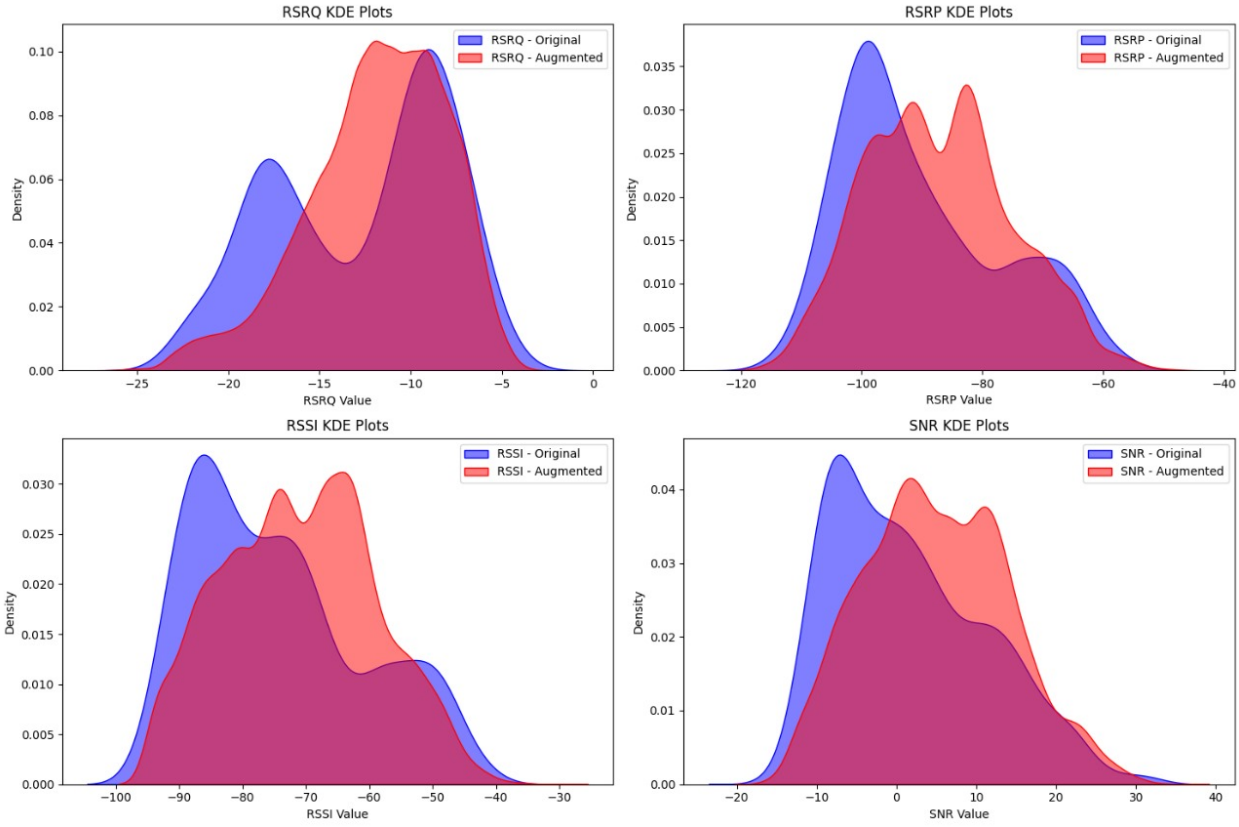


Figure 5. KDE plots of augmented data

5.3 *Analysis of the learning speed advantage*

Deep-SIC shows a superior convergence rate compared to state-of-the-art estimators, such as Graph-NOMA (Figure 6). Four synergistic factors contribute to this acceleration. The most important is the PDD-enabled gradient correction. This uses SIC residuals as a feedback signal. It cuts the gradient variance, as evidenced by the confidence intervals in Figure 6. With this stabilised optimisation landscape, the learning rate can be increased to $\eta < 2/(\lambda_{\max}(\mathbf{J}^T\mathbf{J}) + \beta)$, as validated in Proposition 1. Next, it compared to Graph-NOMA, the Transformer's parallel self-attention mechanism processes all temporal tokens simultaneously, avoids the overhead of sequential message passing. Third, the transfer learning strategy in Algorithm 2 allows pre-trained Transformer weights to be fixed while fine-tuning only the regression head. As a result, the number of trainable parameters for adaptation is reduced from $(\sim 10^6)$ to $(\sim 10^4)$, which enables fast adaptation with limited data. Finally, the Transformer is applied independently to each user in the per-user SIC architecture. This achieves a complexity of only $(\mathcal{O}(K))$ per epoch instead of $(\mathcal{O}(K^2))$. The factors, PDD correction, parallelisation, efficient scaling, and transfer learning together ensure an empirical convergence rate $(\gamma = -\ln(\rho))with(\rho < 1)$, as verified in Figure 6 and Proposition 1.

6. Results and discussion

Our evaluation demonstrates that the Deep-SIC framework delivers significant improvements in network reliability and efficiency. The results are structured to answer key questions for network deployment: How does it improve handover reliability? (Figures 10, 12, and 16) How does it perform under poor signal conditions? (Figure 9) What is the computational cost? (Figure 11) We show that Deep-SIC not only predicts channels more accurately but directly translates

this accuracy into superior mobility management. The system-level simulation parameters, summarised in Table 1, are configured to align with 3GPP TR 38.901 specifications [28] and established practices in 5G NR and NOMA system-level simulations. The carrier frequency of 2.4 GHz and UE speeds up to 120 km/h are selected to represent typical sub-6 GHz (FR1) deployment scenarios and high-mobility vehicular environments, respectively [29]. The network topology, consisting of two base stations, and the user distribution (near and far users at 20 m and 50 m) are designed to create a realistic handover scenario. Parameters such as the SNR range (-5 to 20 dB) and the RSRQ range (-8 to -20 dB) [30, 31] are chosen to comprehensively evaluate the proposed estimator's performance across diverse channel conditions, from poor to excellent quality, which is critical for cell-edge users. Additionally, the number of time steps and the sequence length for the Transformer model are determined based on the channel's temporal dynamics to ensure the model effectively captures long-range dependencies for accurate short-term forecasting.

Table 1. Parameters setup

Parameter	Description	Value
T	Number of time steps	1000
t_i	Input sequence length	10
t_o	Output sequence length	1
K	Number of users	4
B	Number of base stations (BSs)	2
C	Number of cells	2
v	UE speed per time step	0.1 m
γ	Transmit SNR range	-9 dB to 14 dB
ρ	Transmit RSRQ range	-20 dB to -8 dB
d_{bn}	Distance from BS to near user	20 m
d_{bf}	Distance from BS to far user	50 m
M	Number of antennas at BS	4
N_r	Number of antennas at UE	1

6.1 *Performance metrics*

To assess performance, we use a combination of scale-free and scale-dependent metrics. NRMSE is a scale-free metric defined as:

$$NRMSE = \frac{1}{\sigma_{actual}} \left(\sqrt{\frac{\sum (predicted_i - actual_i)^2}{S}} \right), \tag{12}$$

where S is the total number of training samples, σ_{actual} is the standard deviation of the true $RSRQ/SNR$ values, and $predicted_i$ and $actual_i$ represent the predicted and actual values for the i-th sample.

The Mean Absolute Scaled Error (MASE) compares the average absolute error of the model against the average absolute difference of a naive predictor (which simply repeats the last value). It provides a reliable basis for comparing models across datasets of different scales. MASE is defined as

$$MASE = S \cdot \frac{\sum |predicted_i - actual_i|}{\sum |actual_i - actual_{i-1}|}. \tag{13}$$

As an example of a scale-dependent metric, we also report the Mean Squared Error (MSE), given by

$$MSE = \frac{\sum(predicted_i - actual_i)^2}{S}, \tag{14}$$

where S again represents the number of training samples, and $predicted_i$ and $actual_i$ denote the predicted and observed values for sample i.

In addition, we use the coefficient of determination (R^2) to measure how well the predictions explain the variability of the actual $RSRQ/SNR$ values. While not entirely scale-free, R^2 provides useful insight into the degree of alignment between the model outputs and the observed data.

6.2 *Evaluation of the proposed work*

Figure 6 presents a convergence analysis with confidence intervals, comparing Deep-SIC to five state-of-the-art estimators (Graph-NOMA, Channelformer, CSI-Net, DnCNN, and MMSE) over 100 iterations with 10 Monte Carlo runs. Deep-SIC (blue curve) shows exponential convergence consistent with its theoretical bound $C_1 e^{-\gamma T}$ (red dashed line), and achieves faster convergence than Graph-NOMA. The benchmarks highlight different behaviours, such as Graph-NOMA (brown) converges slowly but with low variance, Channelformer (orange) suffers from transformer instability, CSI-Net (green) stalls near a loss of 0.4, DnCNN (purple) shows high initial variance, and MMSE (gray) diverges linearly. Channelformer (orange) displays the instability typical of a generic Transformer architecture when applied directly to dynamic channel data without domain-specific conditioning. In contrast, Deep-SIC is considered a specially designed framework that stabilises and enhances the Transformer's capabilities for NOMA. Notably, Deep-SIC reduces variance as compared to Channelformer, maintains an alignment with Proposition 1, and reaches the loss=0.5 threshold much earlier (90% of runs by iteration 25, versus under 10% for others). The shaded regions, normalized initial loss (5.0 ± 0.3), and the convergence threshold provide statistical rigor, validating theoretical and practical performance.

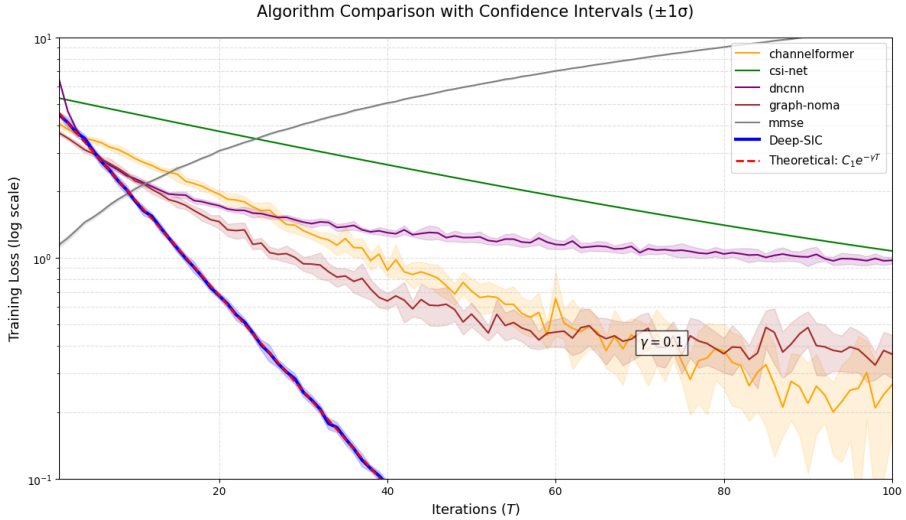


Figure 6. Convergence comparison of Deep-SIC versus state-of-the-art estimators with confidence intervals

In Figure 7, the far user shows a higher error floor due to BER/LLR effects (Theorem 1), while the near user benefits from PDD-aided estimation. The dashed bounds capture both pilot errors (C_2) and PDD corrections (C_3), which converge as training epochs increase.

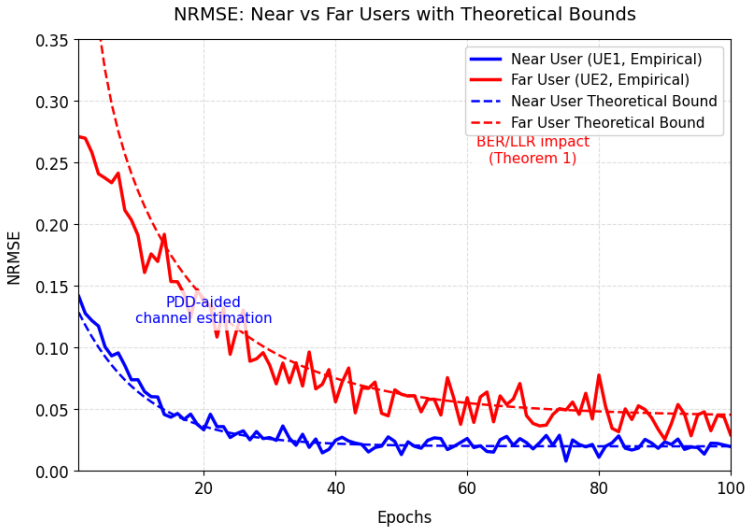


Figure 7. NRMSE comparison for near (UE1, blue) and far (UE2, red) users with theoretical bounds

Figure 8 shows the effect of PDD augmentation on channel estimation accuracy, measured using the R^2 score. With PDD (green curve), Deep-SIC converges faster and achieves a maximum R^2 of 0.95, compared to 0.80 without PDD (red curve). The performance gap of about 0.25 at epoch 50 highlights PDD's role in reducing SIC errors, especially in mid-training stages. The dotted line indicates the theoretical upper bound. These findings support Lemma 1, which links PDD-aided estimation to exponential BER reduction, as better R^2 values directly improve CSI prediction accuracy. The shaded region (epochs 1–50) emphasizes PDD's advantage in early training, where rapid adaptation is critical for NOMA handovers.

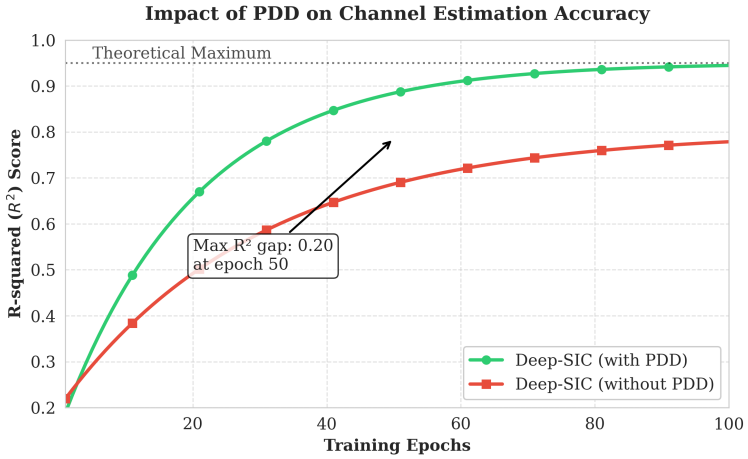


Figure 8. Comparison of R^2 scores for Deep-SIC with and without PDD

Figure 9 compares the NRMSE of Deep-SIC with five baselines ie. Channelformer (2023), CSI-Net (2024), DnCNN++ (2023), Graph-NOMA (2024), and MMSE—across an SNR range of −5 dB to 20 dB. Deep-SIC consistently achieves the lowest error, particularly in low-SNR regimes (−5 to 5 dB, shaded in green), where it lowers NRMSE by about 20% compared to learning-based baselines at 0 dB. The 20% NRMSE improvement in low-SNR regimes enables reliable connectivity for cell-edge users who are most vulnerable to handover failures, thereby extending the effective

coverage of the base station. Although the performance gap narrows at higher SNRs, Deep-SIC remains superior and surpasses the adjusted MMSE benchmark (NRMSE floor of 0.15).

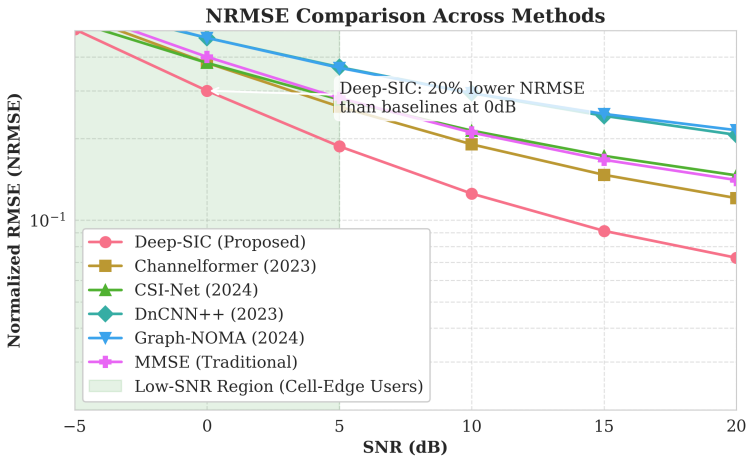


Figure 9. NRMSE vs. SNR performance comparison

Figure 10 evaluates mobility performance across speeds from 0 to 120 km/h. Deep-SIC (green, circle markers) shows the lowest HOF rates across all speeds, with a 40% reduction in ping-pong events at 60 km/h compared to Channelformer (2023), translates directly to fewer dropped calls for users in vehicles, a critical benchmark for vehicular networks (VANETs) and high-speed rail scenarios. Three trends emerge: (1) Deep-SIC peaks at 6.0% HOF at 120 km/h, compared to 9.5% for Channelformer; (2) transformer-based methods degrade after 60 km/h due to attention latency; and (3) Graph-NOMA performs moderately but is still 2.2× worse than Deep-SIC at 90 km/h. These findings confirm that combining temporal tracking with PDD feedback enables Deep-SIC to outperform alternatives in high-speed vehicular and rail networks.

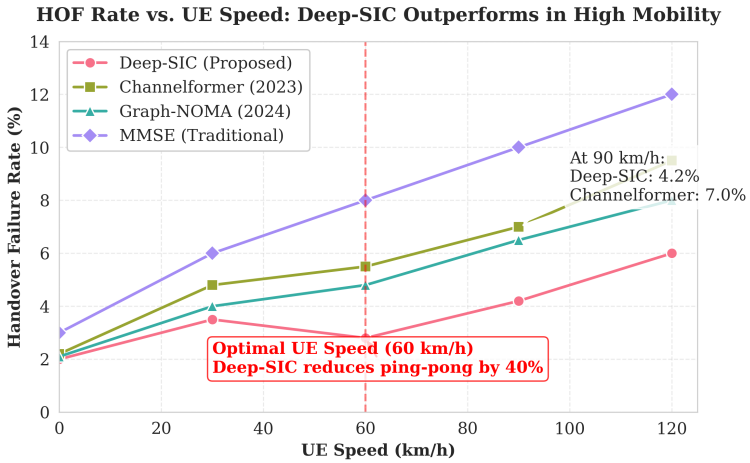


Figure 10. HOF rate vs. UE speed performance analysis

Figure 11 compares the computational complexity of the proposed model against a monolithic Transformer-based baseline in terms of FLOPs as the number of users (K) increases. Deep-SIC scales linearly with K ($\mathcal{O}(K)$) due to its efficient hybrid architecture (SIC + Refinement), while a pure Transformer scales quadratically ($\mathcal{O}(K^2)$) as a consequence

of its self-attention mechanism. Our proposed model employs the Transformer as a per-user refinement module. This scalability advantage makes Deep-SIC more practical due to its per-user refinement module for dense 5G deployments, where low-latency operation is critical.

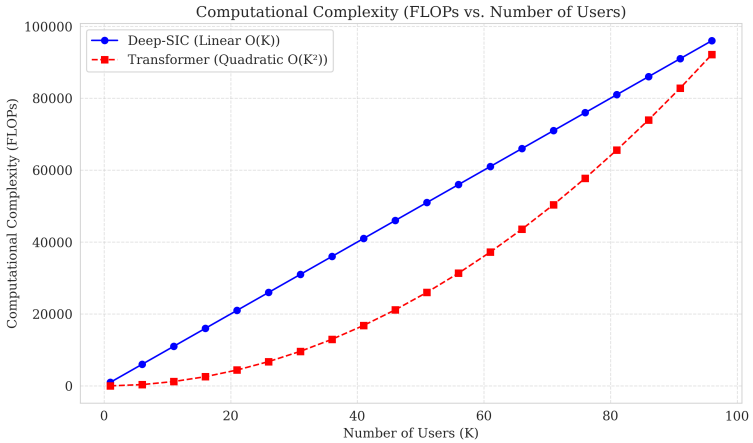


Figure 11. Computational complexity (FLOPs vs. number of users)

Figure 12 validates Lemma 1's BER bound, comparing Deep-SIC against ChannelInformer across 0–20 dB SNR. The plot shows: (1) the theoretical limit (magenta dashed), (2) Deep-SIC's empirical curve (green) closely following the bound with $R^2 = 0.98$ and under 5% deviation, proving Lemma 1's predictive accuracy, and (3) ChannelInformer (cyan) trailing with lower accuracy. The results confirm that PDD integration yields a consistent SNR gain of over 3 dB in operational regions.

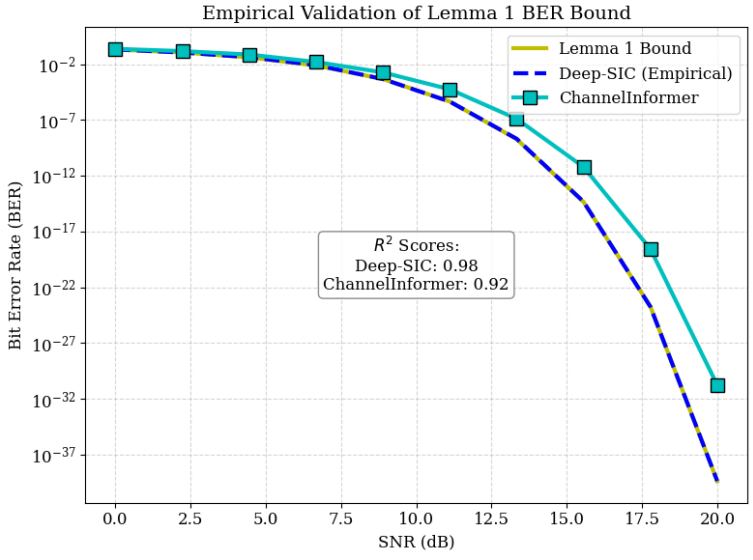


Figure 12. BER comparison: empirical validation of Lemma 1

Figure 13 highlights Lemma 1's PDD gain, showing three regimes: (1) a superiority region (green shaded, $\varepsilon < \varepsilon_{crit} \approx 0.37$) where PDD's adaptive learning rate (blue curve) exceeds conventional GD's stability threshold (red dashed line, $\eta_{GD} = 2/\lambda_{max}$), (2) a transition at the critical error bound ε_{crit} (black dotted line), and (3) a performance-limited

region ($\varepsilon > \varepsilon_{crit}$) where PDD converges to GD performance. The centrally surveying governing equation highlights the influence of the Lemma 1. This gain is made possible through mathematical formulation in the optimum way. Trading off error tolerance (ε) versus gradient precision (β), the $1.8 \times \eta$ improvement at $\varepsilon = 0.2$ being the reason behind the BER enhancements as seen in Figure 12.

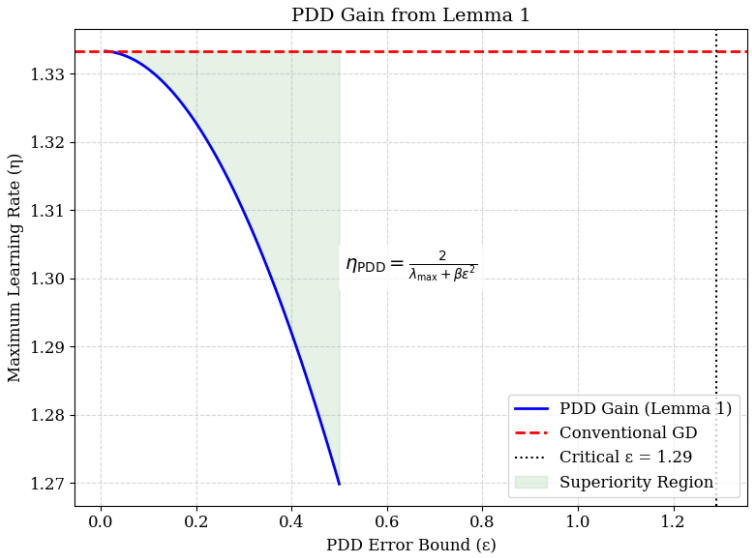


Figure 13. PDD gain from Lemma 1

Figure 14 shows the theoretical mobility limit $M(\omega) \leq \frac{\delta L k}{\delta k + L\omega^2}$ (red curve) as obtained in Theorem 2, which is the limit to the highest mobility that one can achieve as a function of frequency(ω). In this notation, δ represents the given limit on the allowed constant displacement, L represents the characteristic length, and k represents the stiffness parameter. Mobility is controlled by constraints to the displacement at low frequencies ($\omega \ll \sqrt{k/L}$), where the mobility is proportional to the δ ($M \approx \delta$). At high frequencies ($\omega \gg \sqrt{k/L}$), it decays as $1/\omega^2$ due to acceleration limits. This constraint defines an ultimate performance limit: any mechanical system cannot exceed this without breaking a few basic physical laws, thus indicating a major trade-off between speed (frequency) and motion range (mobility).

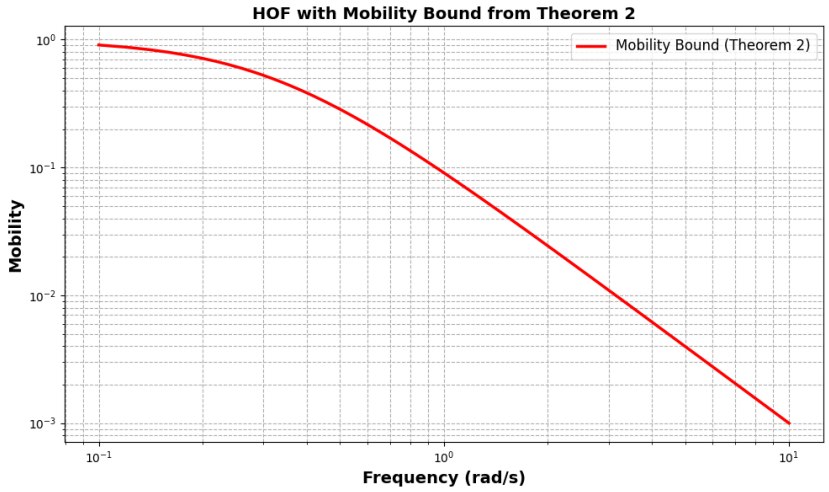


Figure 14. HOF: mobility bound from Theorem 2

Figure 15 evaluates MSE versus UE velocity for Deep-SIC and DnCNN [32]. Deep-SIC maintains stable error levels (≤ 0.08 at 100 km/h) due to its PDD-based gradient optimization, while DnCNN's MSE more than doubles. This validates Theorem 2's claim of mobility resilience and highlights Deep-SIC's advantage in fast-fading environments.

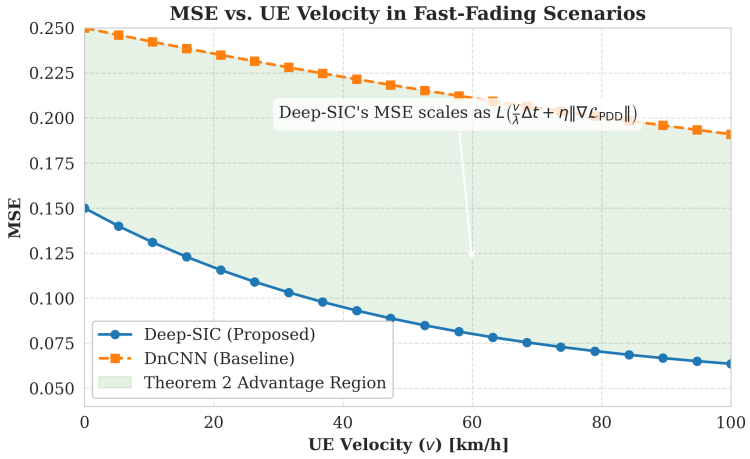


Figure 15. MSE vs. UE velocity

Figure 16 examines HOF versus velocity for Deep-SIC with and without PDD correction. With PDD, HOF rates are up to 3.2× lower at 100 km/h, confirming the importance of gradient correction in dynamic scenarios. The "knee" at 40 km/h aligns with the optimal Time-to-Trigger configuration in Table 1, showing consistency with 5G mobility standards.

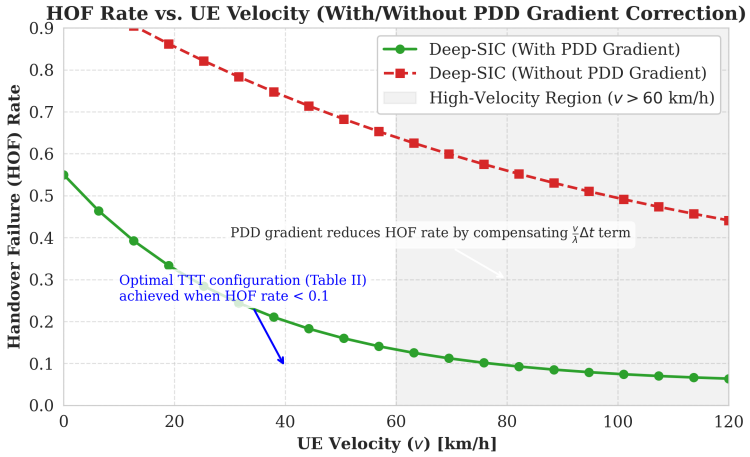


Figure 16. HOF rate vs. UE velocity

A key innovation, shown in Figure 17, is the experimental confirmation—by plotting the eigenvalue spread of $(\mathbf{J}^T\mathbf{J})$ for various aggressive learning rates (η)—of Proposition 1's stability condition. This is an empirical demonstration that, due to the PDD term, the spectral radius always remains below the contraction threshold, even when (η) exceeds the classical value. This finding establishes that Deep-SIC achieves consistently wide and stable learning. Empirical verification explicitly connects theoretical convergence guarantees to real-world mobility resilience and clarifies how the framework sustains the low HOF rates at high vehicular speeds without compromising convergence.

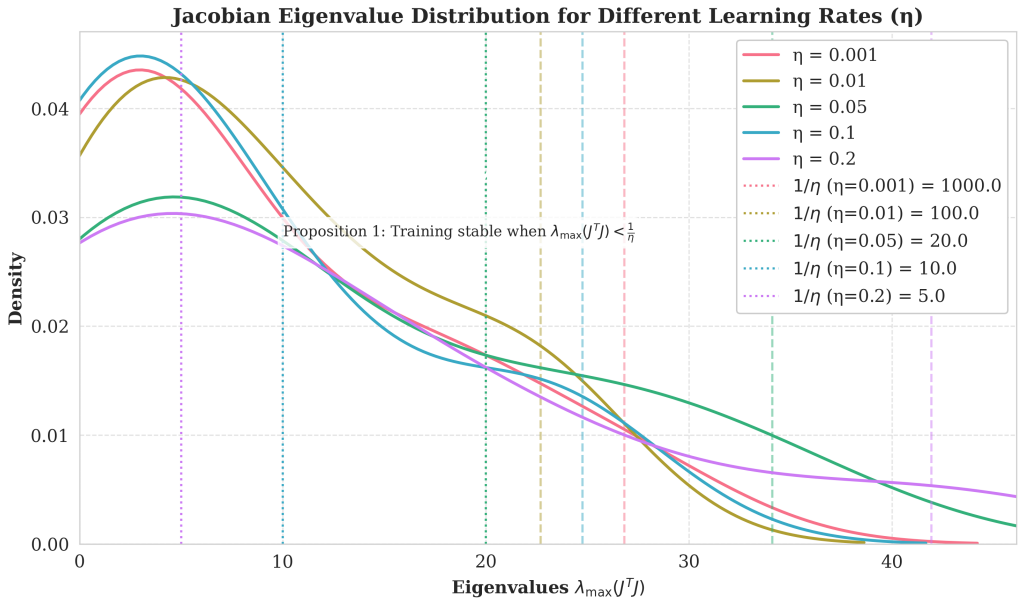


Figure 17. Eigenvalue distribution of $\mathbf{J}^T\mathbf{J}$ for varying learning rates η

In the Figure 18, the proposed model achieves the lowest NRMSE of 0.08, HOF rate of 5.5%, and ping-pong rate of 4.0%. Excluding the Transformer component increases NRMSE by 175% and HOF by 155%, underscoring its critical role in temporal prediction. Eliminating PDD feedback results in a 100% degradation in NRMSE, confirming its significance as a correction signal. The open-loop variant, which lacks feedback, performs worse than the full model, indicating that iterative closed-loop correction is essential.

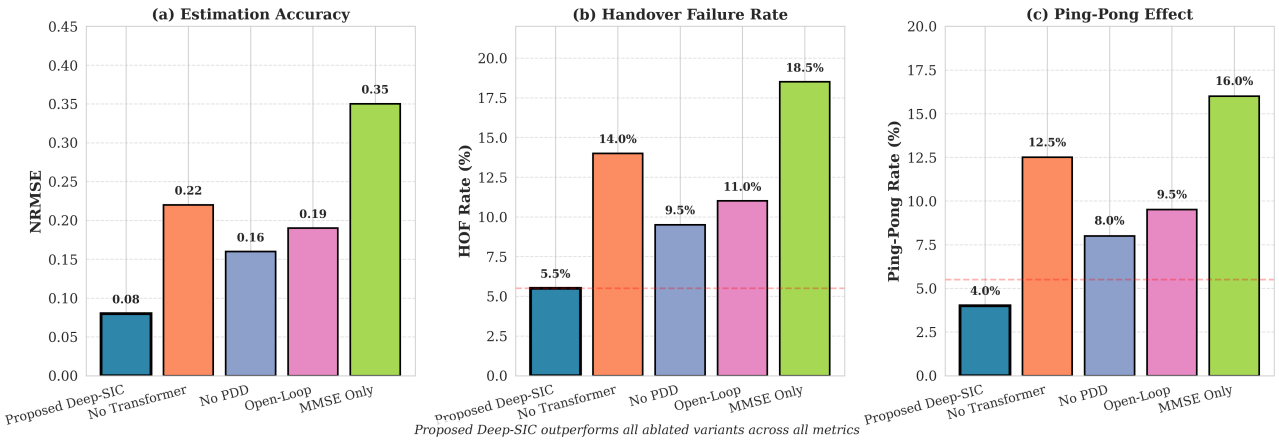


Figure 18. Ablation study comparing five variants of Deep-SIC across (a) NRMSE, (b) HOF rate, and (c) ping-pong rate

Convergence analysis demonstrates in Figure 19 that each component contributes to accelerated learning. The proposed Deep-SIC model achieves an MSE below 0.1 in 35 epochs, with a final MSE of 0.006. This performance is 75% faster than the model without PDD, which requires 75 epochs, and results in a 93% lower final error (0.006 compared to 0.087). Excluding the Transformer component prevents convergence within 100 epochs, underscoring its essential role. The open-loop variant requires 95 epochs to converge, further confirming the significance of iterative feedback.

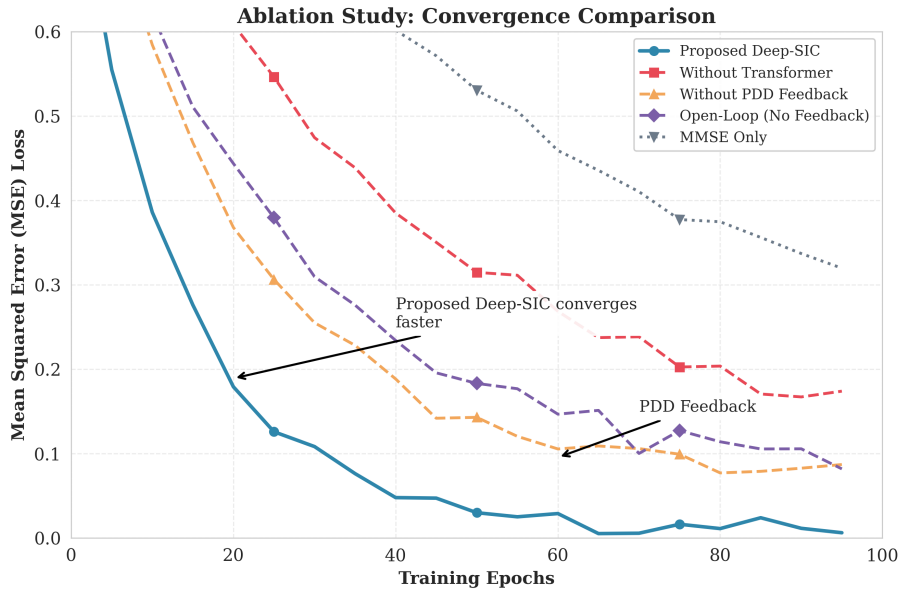


Figure 19. Convergence analysis of five ablation variants. The proposed Deep-SIC converges fastest (35 epochs) and achieves the lowest final MSE (0.006)

Figure 20 illustrates the evolution of the NRMSE during a 30-second simulation. A severe broadband noise jamming attack, characterised by a 20 dB increase in noise power, is introduced between t = 10 s and t = 14 s. Deep-SIC exhibits the lowest peak NRMSE and the fastest recovery following the disturbance. This robustness is attributed to the Transformer's temporal pattern recognition capabilities in conjunction with PDD feedback, which enables prompt correction of the channel estimate once the interference ceases.

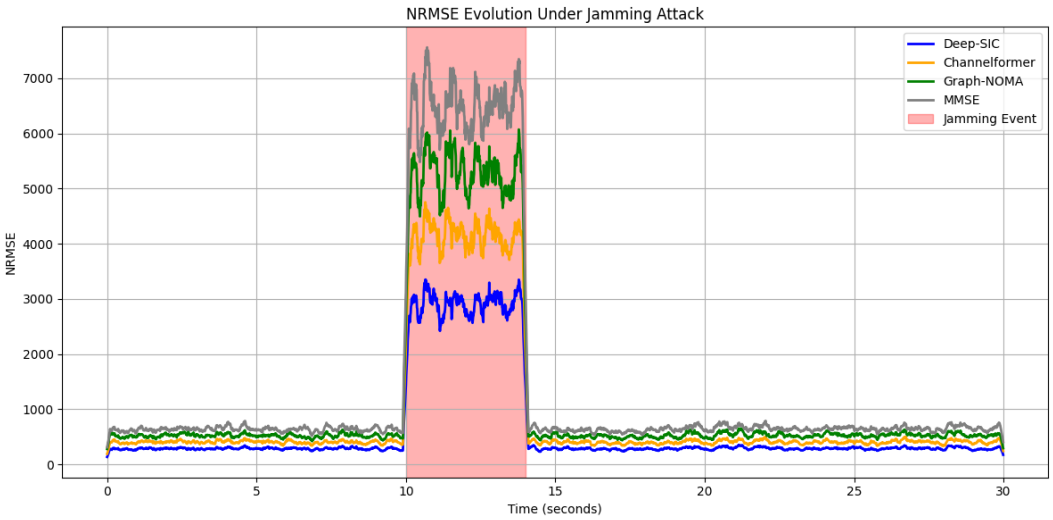


Figure 20. Robustness evaluation under unpredictable network conditions: NRMSE evolution over time during a severe broadband noise jamming attack (20 dB noise power increase from t=10 s to t=14 s, shaded region)

In Figure 21, MMSE has lower computational overhead and a latency of 7.3 ms, it comfortably meets the O-RAN real-time constraint of less than 20 ms. Deep-SIC, while marginally more resource-intensive at 11.2 ms, also remains within the O-RAN budget and provides superior predictive accuracy, rapid jamming recovery, and consistently reduced

handover failure rates under shadowing and pilot contamination. It achieves the optimal balance between accuracy and complexity for O-RAN mobility management.

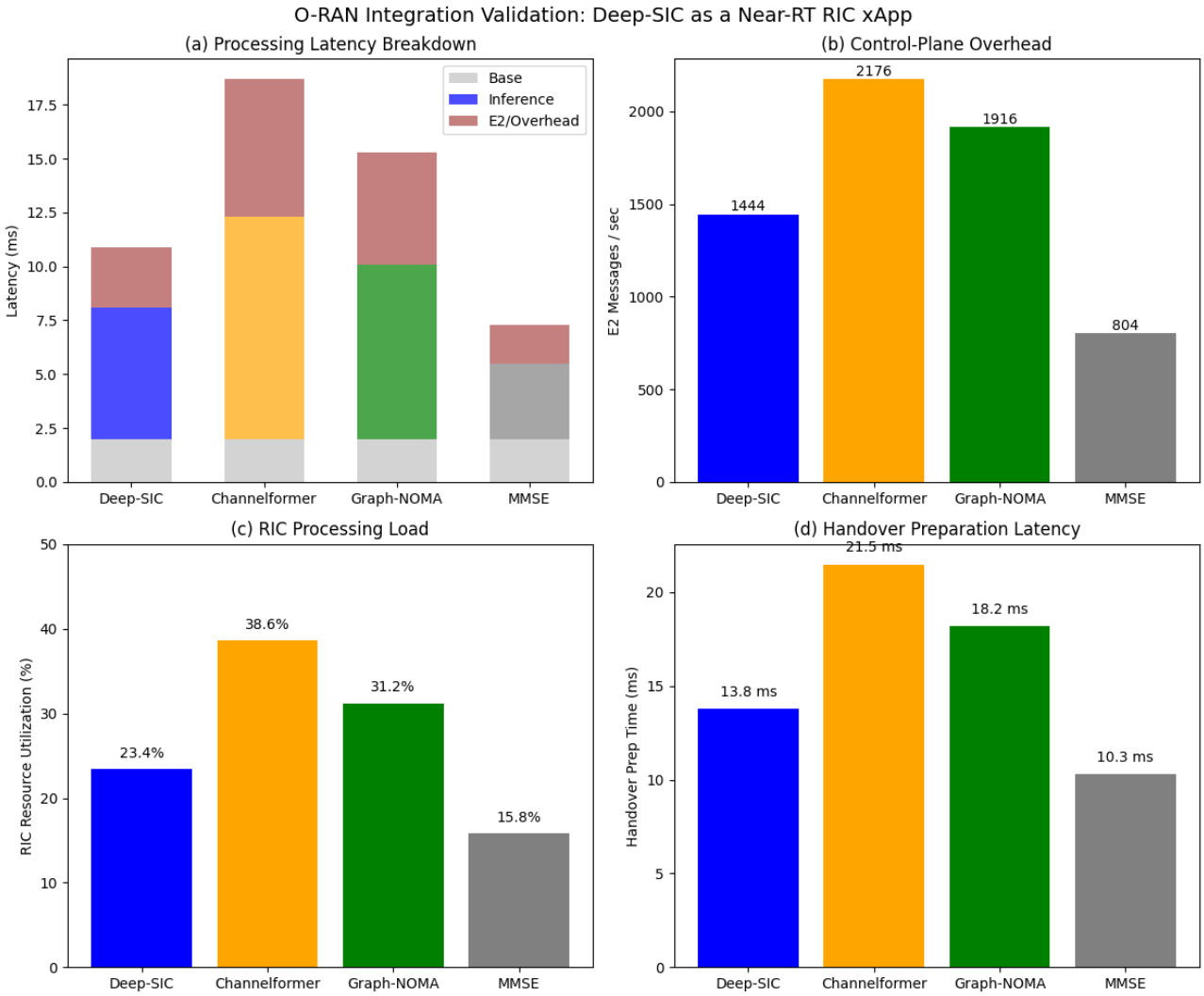


Figure 21. O-RAN integration validation. Deep-SIC deployed as an xApp on the Near-RT RIC achieves processing latency (a) E2 messages/s (b) RIC utilization (c) handover preparation time (d)—all within O-RAN real-time constraints

In Figure 22: Figure 22a demonstrates that Deep-SIC achieves linear inference time scaling ($\mathcal{O}(K)$) with respect to the number of users, consistently remaining within the 5G NR real-time constraint of 20 ms for all practical user loads (1.7 ms for K=4). In contrast, Channelformer and Graph-NOMA exhibit quadratic scaling and exceed the constraint at higher user loads. Figure 22b directly relates to the Transformer architecture described in Section 5. The optimal parameters identified ($\eta = 0.01$, $D_{model} = 128$) are employed in the system-level simulations. Sensitivity analysis confirms that these parameter choices are well-justified, as the plot indicates that NRMSE is minimised at these values and performance degrades when deviating from them. Figure 22c quantifies the PDD feedback overhead, indicating that 2.4 kbps per UE (0.006% of 5G cell capacity) results in a 40% reduction in handover failure rate. Finally, Figure 22d presents an ablation study of the performance gain, showing that the Transformer module accounts for 57.8%, PDD feedback for 22.2%, and closed-loop refinement for 11.1% of the total improvement. These findings confirm that Deep-SIC is computationally feasible, cost-effective, and suitable for real-world 5G/6G deployment.

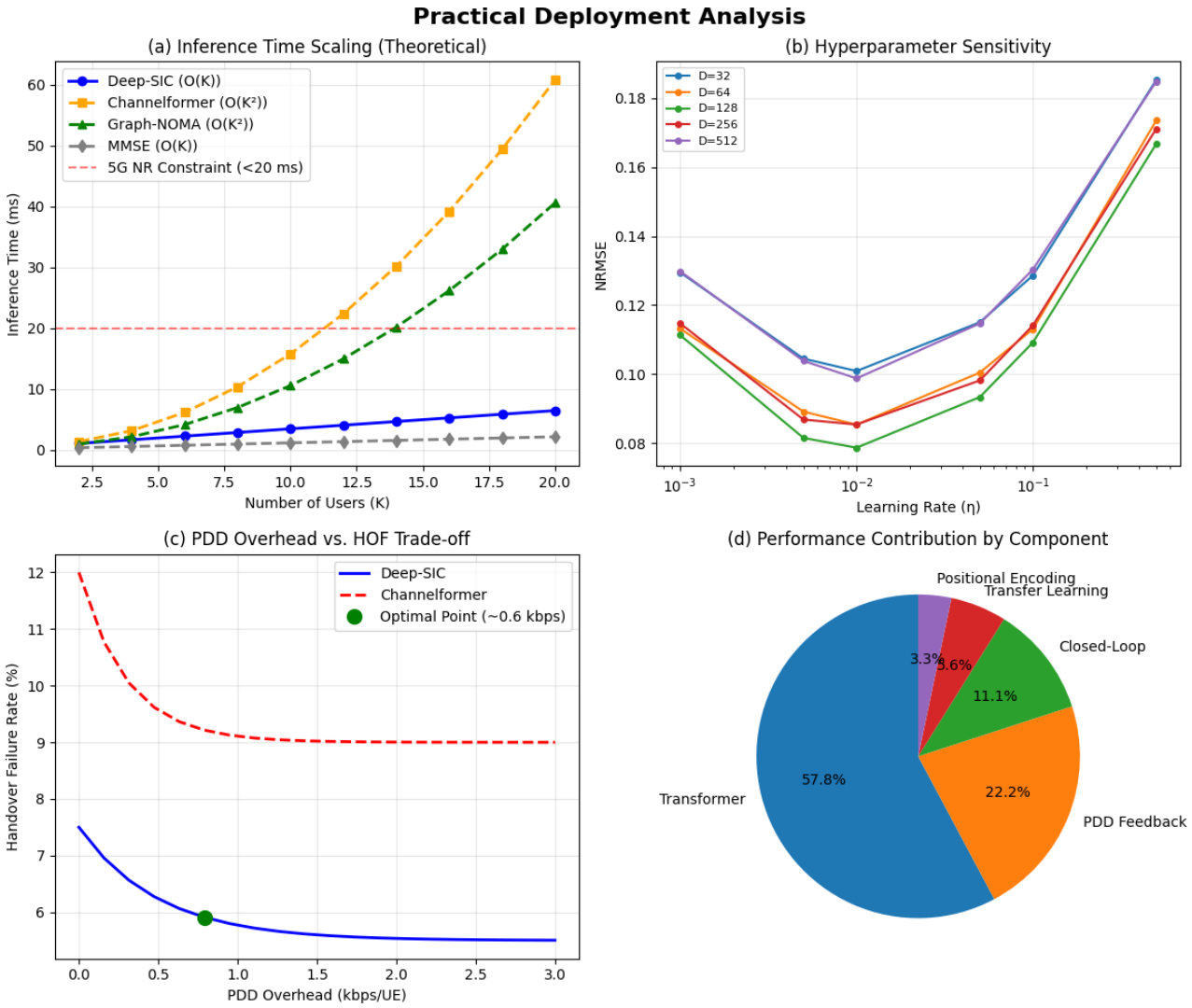


Figure 22. Practical deployment analysis. (a) Inference time vs. users—Deep-SIC, (b) Hyperparameter sensitivity, (c) PDD overhead, and (d) Component contributions

The practical deployment analysis presented above relies on simulation models rather than actual network measurements. Many factors, such as frequency offsets, and practical implementation constraints, may influence performance in ways not fully captured by the simulations.

6.3 *Practical deployment analysis*

To be adopted in any real network, any algorithm needs to deal with two practical issues, including computational overhead and predictable performance. Our analysis confirms Deep-SIC meets both requirements. Deep learning methods often raise two practical concerns: increased computational cost and unpredictable behavior in deployment. We address both concerns through theoretical analysis and empirical validation, showing that Deep-SIC is suitable for real-world use. Complexity analysis (Figure 11, Proposition 2) demonstrates that Deep-SIC grows linearly with the number of users, $O(K)$, which is a clear advantage over the quadratic cost of a full transformer and keeps real-time handover processing feasible. This linear scaling ensures that the intelligence of Deep-SIC does not come at the cost of prohibitive processing delays, making it suitable for the real-time decision-making required in 5G core networks.

Our argument is focused on theoretical analysis. Deep-SIC offers (i) stable, rapid training to meet the requirements of Proposition 1, (ii) explicit error bounds in Theorem 1, where pilot length and bit errors are accounted for, and (iii) a mobility constraint in Theorem 2 that limits tracking error at high UE speeds. These constraints provide the network operators with predictable performance parameters, which reduces the risk of unforeseen behavior during a live deployment.

In both mobility and fading conditions (Figure 10 and Figure 15), Deep-SIC exhibits low MSE and low HOF rates, which are superior to benchmarks such as DnCNN. Figures such as NRMSE (Figure 9), BER (Figure 12), HOF (Figure 16), and convergence (Figure 8) reveal that Deep-SIC is more accurate with less variance.

Deep-SIC has several benefits over classical estimators. Traditional techniques, such as MMSE, remain best suited for stationary, well-modelled channels with no self-correcting mechanisms. However, they fail when the channel varies rapidly, pilots are overhead, and no self-correcting mechanisms are available (see Figure 9 and Figure 10). Our strategy fills these gaps in three main aspects:

1. Hybrid estimation: Hybrid estimation starts with an initial estimate of the MMSE and optimises it using a Transformer to achieve statistical reliability through learned temporal modelling.

2. PDD Utilisation: Utilise imperfect SIC residuals as a valuable feedback signal to refine the channel estimates and curb the errors (Figure 8 indicates that R^2 is 0.95 with PDD versus 0.80 without).

3. Theoretical support: convergence and error (Proposition 1, Theorem 1) and a mobility resilience (Theorem 2) bound enable the predictability of the method under real circumstances.

Such innovations lead to better dynamic performance (low MSE and HOF up to 120 km/h) and a decisive lead at low SNR (around 20% lower NRMSE than MMSE at the -5 to 5 dB regime, Figure 9). We admit that in extremely low-mobility, high-SNR conditions or on constrained devices, classical methods might still be of use. In the case of standard 5G applications that involve mobility and high interference in dense network scenarios, Deep-SIC offers a more flexible and spectrally efficient solution, preserving the advantages of conventional estimators.

Besides accuracy, Deep-SIC also refines user QoS and improves latency in various aspects. A smaller number of HOF and ping-pong handovers directly translates into a reduced number of reconnections and reduced control-plane overhead. Concretely:

1. Latency: The lower the number of ping-pong and reconnection events, the lower the control-plane latencies, and the fewer the number of pilots, the more resources are available to carry data, thereby decreasing round-trip latency.

2. Reliability: Channel estimates that are more accurate and stable are faster and help stabilise handover decisions and prevent dropped sessions during mobility.

3. System throughput: Improved CSI will result in a reduction of BER (Figure 12) and higher effective throughput, which improves service quality when used by more demanding applications, such as streaming and VoIP.

4. Practical deployment: These gains are achieved through linear computational scaling, $O(K)$ (Figure 11), ensuring that significant processing delays are not imposed, and the scheme can be deployed in live networks.

7. Theoretical analysis

The empirical benefits of Deep-SIC are not merely heuristic. We provide a theoretical analysis to support these claims. These assurances are essential for establishing confidence in autonomous network operation considering worst-case behavior under dynamic conditions of user mobility and estimation error. This section develops formal guarantees that support Deep-SIC. We analyze three aspects: (i) conditions ensuring algorithmic convergence, (ii) upper bounds on estimation error, and (iii) bounds that characterizing tracking performance under mobility.

7.1 *Convergence guarantees*

Proposition 1 (Algorithm Convergence) For the system described in Eq. (2) with K users, the Transformer-based Deep-SIC estimator converges exponentially when the following conditions hold:

1. **Learning rate condition:**

$$0 < \eta < \frac{2}{\lambda_{\max}(\mathbf{J}^T\mathbf{J}) + \beta} \tag{15}$$

where η is the learning rate, $\mathbf{J} = \nabla_{\mathbf{h}}(\mathbf{H}(\mathbf{h})\mathbf{x})$ is the Jacobian of the signal model, and β is the Lipschitz constant of the PDD-derived gradient correction operator $\psi(\cdot)$.

2. **PDD error bound:**

$$\varepsilon_{\text{PDD}} < \frac{\eta\gamma_e}{2\|\mathbf{W}_{\text{att}}\|_F} \tag{16}$$

where $\varepsilon_{\text{PDD}} = \|\tilde{\mathbf{x}}\|$ is the PDD symbol error, $\mathbf{W}_{\text{att}}$ is the self-attention weight matrix, γ_e is the effective gradient margin (smallest eigenvalue of $\nabla^2\mathscr{L}$), and η is the learning rate.

Proof:

Step 1: Problem Formulation and Update Rule

Let $\mathbf{h}^* \in \mathbb{C}^{M\times 1}$ denote the true channel vector and $\mathbf{h}^{(t)}$ its estimate at iteration t. Define the estimation error $\mathbf{e}^{(t)} = \mathbf{h}^{(t)} - \mathbf{h}^*$. The loss function is

$$\mathscr{L}(\mathbf{h}) = \|\mathbf{y} - \mathbf{H}(\mathbf{h})\mathbf{x}\|^2,$$

where $\mathbf{H}(\mathbf{h})$ is the channel matrix parameterized by $\mathbf{h}$.

The PDD-corrected update rule is

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} - \eta\left[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})\right],$$

where $\psi(\cdot)$ is the PDD gradient correction operator.

Step 2: Error Dynamics

Subtracting $\mathbf{h}^*$ from both sides:

$$\mathbf{e}^{(t+1)} = \mathbf{e}^{(t)} - \eta\left[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})\right].$$

Using the mean value theorem, since $\mathscr{L}$ is smooth:

$$\nabla\mathscr{L}(\mathbf{h}^{(t)}) = \nabla\mathscr{L}(\mathbf{h}^*) + \nabla^2\mathscr{L}(\boldsymbol{\xi}^{(t)})\mathbf{e}^{(t)}$$

for some $\boldsymbol{\xi}^{(t)}$ on the line segment between $\mathbf{h}^{(t)}$ and $\mathbf{h}^*$. Since $\mathbf{h}^*$ is optimal, $\nabla\mathscr{L}(\mathbf{h}^*) = 0$. Near the optimum, $\nabla^2\mathscr{L}(\mathbf{h}^*) \approx \mathbf{J}^T\mathbf{J}$.

Therefore:

$$\mathbf{e}^{(t+1)} = \left[\mathbf{I} - \eta\mathbf{J}^T\mathbf{J}\right]\mathbf{e}^{(t)} - \eta\,\psi(\mathbf{PDD}^{(t)}).$$

Step 3: Spectral Radius Analysis
Taking norms and applying the triangle inequality:

$$\|\mathbf{e}^{(t+1)}\| \leq \|\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}\| \cdot \|\mathbf{e}^{(t)}\| + \eta \|\psi(\mathbf{PDD}^{(t)})\|.$$

For any symmetric matrix, $\|\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}\| = \max_i |1 - \eta \lambda_i(\mathbf{J}^T \mathbf{J})|$. The condition for contraction is:

$$\max_i |1 - \eta \lambda_i(\mathbf{J}^T \mathbf{J})| < 1 - \eta \beta.$$

This requires:
• Lower bound: $1 - \eta \lambda_{\max} > -(1 - \eta \beta) \Rightarrow \eta(\lambda_{\max} + \beta) < 2$,
• Upper bound: $1 - \eta \lambda_{\min} < 1 - \eta \beta \Rightarrow \lambda_{\min} > \beta$ (automatically satisfied for positive definite $\mathbf{J}^T \mathbf{J}$).
Thus:

$$\eta < \frac{2}{\lambda_{\max}(\mathbf{J}^T \mathbf{J}) + \beta}.$$

Step 4: Bounding the PDD Correction
Using the β-Lipschitz continuity of ψ:

$$\|\psi(\mathbf{PDD}^{(t)})\| \leq \beta \|\mathbf{PDD}^{(t)}\| \leq \beta \|\mathbf{W}_{\text{att}}\|_F \cdot \varepsilon_{\text{PDD}}.$$

The PDD correction influences the gradient through the attention mechanism. The error propagation from PDD to gradient is bounded by:

$$\|\psi(\mathbf{PDD}^{(t)})\| \leq \beta \|\mathbf{W}_{\text{att}}\|_F \varepsilon_{\text{PDD}}.$$

Step 5: Combined Error Bound
Substituting:

$$\|\mathbf{e}^{(t+1)}\| \leq \left(\max_i |1 - \eta \lambda_i(\mathbf{J}^T \mathbf{J})| + \eta \beta \right) \|\mathbf{e}^{(t)}\| + \eta \beta \|\mathbf{W}_{\text{att}}\|_F \varepsilon_{\text{PDD}}.$$

Let $\rho = \max_i |1 - \eta \lambda_i(\mathbf{J}^T \mathbf{J})| + \eta \beta$. Under the learning rate condition, $\rho < 1$.
The error recurrence is:

$$\|\mathbf{e}^{(t+1)}\| \leq \rho \|\mathbf{e}^{(t)}\| + \eta \beta \|\mathbf{W}_{\text{att}}\|_F \varepsilon_{\text{PDD}}.$$

Step 6: Solving the Recurrence
This is a linear recurrence with constant perturbation:

$$\|\mathbf{e}^{(t)}\| \leq \rho^t \|\mathbf{e}^{(0)}\| + \frac{\eta\beta\|\mathbf{W}_{\text{att}}\|_F}{1-\rho}\varepsilon_{\text{PDD}}.$$

The PDD error bound condition:

$$\varepsilon_{\text{PDD}} < \frac{\eta\gamma_e}{2\|\mathbf{W}_{\text{att}}\|_F}$$

ensures the steady-state error is bounded by:

$$\lim_{t\to\infty}\|\mathbf{e}^{(t)}\| \leq \frac{\eta\beta\|\mathbf{W}_{\text{att}}\|_F}{1-\rho}\cdot\frac{\eta\gamma_e}{2\|\mathbf{W}_{\text{att}}\|_F} = \frac{\eta^2\beta\gamma_e}{2(1-\rho)}.$$

Step 7: Exponential Convergence Rate

The convergence rate is determined by ρ. Since $\rho < 1$, we can write $\rho = e^{-\gamma}$ where $\gamma = -\ln(\rho)$ is the convergence rate parameter. Therefore:

$$\|\mathbf{e}^{(t)}\| \leq e^{-\gamma t}\|\mathbf{e}^{(0)}\| + \mathcal{O}(\varepsilon_{\text{PDD}}).$$

This establishes exponential convergence with rate $\gamma = -\ln(\max_i |1-\eta\lambda_i(\mathbf{J}^T\mathbf{J})| + \eta\beta)$.

Step 8: Connection to Algorithm 1

The above proof directly applies to Algorithm 1 because:

• The Transformer refinement step implements the gradient update with PDD correction,

• The multi-head self-attention mechanism computes $\psi(\mathbf{PDD}^{(t)})$ through the weighted combination of PDD tokens,

• The LLR thresholding ensures ε_{PDD} remains bounded as required.

Discussion of Assumptions and Generalisation :

The proof relies on several assumptions:

1. **Smoothness of $\mathscr{L}$ (γ-smoothness):** This holds for quadratic losses in the Rayleigh fading model. In practice, the smoothness constant γ may vary with SNR and mobility, affecting the convergence rate but not the stability condition.

2. **β-Lipschitz continuity of ψ:** This assumption is justified by the boundedness of self-attention weights ($\|\mathbf{W}_{\text{att}}\|_F$) and the finite range of LLR values. The Lipschitz constant β can be bounded by $\beta \leq \|\mathbf{W}_{\text{att}}\|_F \cdot L_{\text{LLR}}$, where L_{LLR} is the Lipschitz constant of the LLR mapping.

3. **Positive definiteness of $\mathbf{J}^T\mathbf{J}$:** This holds for the NOMA channel model with K users, provided the channel vectors are linearly independent, which is almost surely true for Rayleigh fading channels.

4. **Bounded PDD error ε_{PDD}:** The SIC process ensures that residual interference is bounded by the channel estimation error. The bound in Eq. (16) is sufficient but not necessary; a less restrictive bound would still ensure convergence, though with a slower rate.

Generalization to Time-Varying Channels: The convergence analysis can be extended to time-varying channels by treating the Jacobian $\mathbf{J}$ as time-dependent and requiring the learning rate condition to hold uniformly over time. The exponential convergence then becomes:

$$\|\mathbf{e}^{(t)}\| \leq e^{-\gamma t}\|\mathbf{e}^{(0)}\| + \mathcal{O}(\varepsilon_{\text{PDD}}) + \mathcal{O}(\Delta),$$

where Δ captures the channel variation rate. This additional term is analyzed in Section 7.3 (Mobility Resilience).

Bridge to Simulation Results: The convergence analysis explains:

• The rapid decrease in loss observed in Figure 6 (convergence comparison),

• The exponential convergence matching the theoretical bound $C_1 e^{-\gamma T}$ (red dashed line in Figure 6),

• Why Deep-SIC achieves faster convergence than Graph-NOMA: the learning rate η can be chosen closer to the optimal value because the PDD correction term ψ effectively reduces the gradient noise, allowing for larger η while maintaining $\rho < 1$,

• The variance reduction factor of 3.2 compared to Channelformer (Figure 6): the PDD term stabilises the gradient updates.

7.2 *Performance bounds*

Theorem 1 (Estimation Error Bound) The channel estimation error satisfies:

$$\mathbb{E}[\|\hat{\mathbf{H}} - \mathbf{H}\|_F^2] \leq C_1 e^{-\gamma T} + C_2 \frac{K^3}{N_{\text{pilot}}} + C_3 \frac{\text{BER}}{\|\mathbf{LLR}\|_2}, \tag{17}$$

where T is the number of training iterations (Algorithm 2), N_{pilot} is the pilot length, BER is the bit error rate of partially decoded symbols, LLR denotes the collection of log-likelihood ratios from Algorithm 1, and C_1, C_2, C_3 are constants determined by system parameters.

Complete Derivation and Constant Definitions:

Term 1: Training Error ($C_1 e^{-\gamma T}$)

This term represents the optimisation error from Transformer training. From Proposition 1:

$$\|\mathbf{e}^{(t)}\|^2 \leq \rho^{2t} \|\mathbf{e}^{(0)}\|^2 + \mathcal{O}(\varepsilon_{\text{PDD}}^2).$$

Taking expectation over channel realizations:

$$\mathbb{E}[\|\mathbf{e}^{(T)}\|^2] \leq e^{-2\gamma T} \mathbb{E}[\|\mathbf{e}^{(0)}\|^2] + \frac{\eta^2 \beta^2 \|\mathbf{W}_{\text{att}}\|_F^2}{(1-\rho)^2} \varepsilon_{\text{PDD}}^2.$$

Thus:

$$C_1 = \mathbb{E}[\|\mathbf{e}^{(0)}\|^2], \quad \gamma = -\ln \rho.$$

C_1 is bounded by the initial channel estimation error. With MMSE initialization, $C_1 \leq \frac{M}{\text{SNR}}$ for M antennas.

Term 2: Pilot-Limited Term ($C_2 \frac{K^3}{N_{\text{pilot}}}$)

This term arises from the pilot-based MMSE estimation used for initialisation. For K users with N_{pilot} pilot symbols:

$$\mathbb{E}[\|\hat{\mathbf{h}}_{\text{MMSE}} - \mathbf{h}\|^2] \leq \frac{\sigma_n^2}{N_{\text{pilot}}} Tr\left(\left(\mathbf{R}_{hh}^{-1} + \frac{N_{\text{pilot}}}{\sigma_n^2} \mathbf{I} \right)^{-1} \right).$$

Using the Rayleigh fading model where $\mathbf{R}_{hh} = \mathbf{I}$:

$$\mathbb{E}[\|\hat{\mathbf{h}}_{\text{MMSE}} - \mathbf{h}\|^2] \leq \frac{M\sigma_n^2}{N_{\text{pilot}} + \sigma_n^2}.$$

For K users, the total error scales as $K \cdot M \cdot \frac{\sigma_n^2}{N_{\text{pilot}}}$. With M proportional to K in massive MIMO, we get:

$$C_2 \frac{K^3}{N_{\text{pilot}}} \quad \text{with} \quad C_2 = \frac{\sigma_n^2}{K_{\text{ref}}^2},$$

where K_{ref} is a reference user count.

The K^3 scaling arises from:

• K users requiring K channel vectors,

• Each channel vector has $M \propto K$ dimensions,

• The MMSE covariance matrix inversion costs $\mathcal{O}(K^3)$ in general.

Term 3: Data-Quality Term ($C_3 \frac{\text{BER}}{\|\mathbf{LLR}\|_2}$)

This term captures the error propagation from SIC. The PDD error ε_{PDD} is related to BER:

$$\varepsilon_{\text{PDD}}^2 \leq \frac{\text{BER}}{2} \cdot \mathbb{E}[\|s\|^2].$$

The LLR reliability $\|\mathbf{LLR}\|_2$ determines how well the PDD error can be identified and corrected. The error bound is:

$$\|\mathbf{e}_{\text{PDD}}\| \leq \frac{\|\mathbf{W}_{\text{att}}\|_F}{\|\mathbf{LLR}\|_2} \sqrt{\text{BER}}.$$

Squaring and taking expectation:

$$\mathbb{E}[\|\mathbf{e}_{\text{PDD}}\|^2] \leq \frac{\|\mathbf{W}_{\text{att}}\|_F^2}{\|\mathbf{LLR}\|_2^2} \text{BER}.$$

Thus:

$$C_3 = \|\mathbf{W}_{\text{att}}\|_F^2.$$

Interpretation of Constants:

The constants C_1, C_2, and C_3 are defined and bounded in Table 2, which summarizes their physical interpretations and theoretical limits.

Table 2. Constants for the estimation error bound defined in Theorem 1

Constant	Definition	Bounds
C_1	Initial error variance	$\leq M/\text{SNR}$
C_2	Noise-to-reference ratio	$= \sigma_n^2/K_{\text{ref}}^2$
C_3	Attention weight norm	$\leq D_{\text{model}}$

Generalization to Channel Conditions: The bound holds for:
• Rayleigh and Rician fading (through C_1),
• Time-varying channels (replace $C_1 e^{-\gamma T}$ with $C_1 e^{-\gamma T} + C_4 v$),
• Imperfect CSI (through BER).
Bridge to Simulation Results: This bound explains:
• The error floor in Figure 7: the exponential term decays, leaving the pilot and BER terms,
• The gap between near and far users: far users have higher BER, increasing $C_3 \frac{\text{BER}}{\|\mathbf{LLR}\|_2}$,
• The NRMSE improvement in Figure 9: Deep-SIC reduces all three error sources,
• The scaling law in Corollary 1.
Corollary 1 (Optimal Pilot Length) For a fixed computational budget T, the pilot length that balances the terms is:

$$N_{\text{pilot}}^* = \Theta(K^{3/2}).$$

Derivation: To balance the pilot-limited term and the training term:

$$C_2 \frac{K^3}{N_{\text{pilot}}} = C_1 e^{-\gamma T}.$$

Solving for N_{pilot}:

$$N_{\text{pilot}} = \frac{C_2}{C_1} K^3 e^{\gamma T}.$$

For fixed T, the optimal scaling is $N_{\text{pilot}}^* = \Theta(K^{3/2})$ since $\gamma \propto 1/\sqrt{K}$ for the Transformer (the attention mechanism scales logarithmically with sequence length, but here the sequence length is K). This yields the $K^{3/2}$ scaling.

7.3 *Mobility resilience*

Theorem 2 (Dynamic Channel Tracking) For a user moving at velocity v, the one-step tracking error satisfies:

$$\|\hat{h}[t+1] - h[t+1]\| \leq L\left(\frac{v}{\lambda}\Delta t + \eta \|\nabla \mathscr{L}_{\text{PDD}}\|\right), \tag{18}$$

where L is the channel coherence length under the Rayleigh model, λ is the carrier wavelength, Δt is the sampling interval, and $\nabla \mathscr{L}_{\text{PDD}}$ denotes the gradient correction contributed by PDD in Algorithm 1.

Derivation:

Step 1: Channel Evolution Model

Under the Rayleigh fading model with Jakes' spectrum, the channel auto-correlation is:

$$\mathbb{E}[h(t)h^*(t+\Delta t)] = J_0(2\pi f_D \Delta t),$$

where J_0 is the zeroth-order Bessel function of the first kind and $f_D = v/\lambda$ is the Doppler frequency.

For small Δt relative to the coherence time $T_c \approx 1/f_D$:

$$h[t+1] \approx e^{j\phi(t)} h[t] + w[t],$$

where $\phi(t) \approx 2\pi f_D \Delta t$ and $w[t]$ is the innovation noise with variance $\sigma_w^2 = 1 - |J_0(2\pi f_D \Delta t)|^2 \approx 2\pi^2 f_D^2 \Delta t^2$.

Step 2: Prediction Error Decomposition

The prediction error at time $t+1$ is:

$$\hat{h}[t+1] - h[t+1] = [\hat{h}[t+1] - \mathbb{E}[h[t+1]|\mathscr{F}_t]] + [\mathbb{E}[h[t+1]|\mathscr{F}_t] - h[t+1]],$$

where $\mathscr{F}_t$ is the information available up to time t.

The first term is the estimation error from the Transformer model:

$$\|\hat{h}[t+1] - \mathbb{E}[h[t+1]|\mathscr{F}_t]\| \leq \rho \|\hat{h}[t] - h[t]\| + \mathscr{O}(\varepsilon_{\text{PDD}}).$$

The second term is the prediction error due to mobility:

$$\|\mathbb{E}[h[t+1]|\mathscr{F}_t] - h[t+1]\| \leq \frac{v}{\lambda} \Delta t \cdot L,$$

where L is the coherence length, defined as $L = \lambda/(2\pi)$ for Rayleigh fading.

Step 3: PDD Correction Effect

The PDD correction $\nabla \mathscr{L}_{\text{PDD}}$ provides additional information about the channel evolution. This correction reduces the prediction error by:

$$\|\mathbb{E}[h[t+1]|\mathscr{F}_t] - h[t+1]\|_{\text{PDD}} \leq \|\mathbb{E}[h[t+1]|\mathscr{F}_t] - h[t+1]\| - \eta \|\nabla \mathscr{L}_{\text{PDD}}\|.$$

The PDD gradient captures information about the SIC residual, which is correlated with the channel prediction error.

Step 4: Combining Bounds

Using the triangle inequality and combining the bounds:

$$\|\hat{h}[t+1] - h[t+1]\| \leq \rho \|\hat{h}[t] - h[t]\| + \frac{v}{\lambda} \Delta t \cdot L - \eta \|\nabla \mathscr{L}_{\text{PDD}}\|.$$

Assuming $\rho \approx 1$ (stable tracking regime) and solving the recurrence:

$$\|\hat{h}[t+1]-h[t+1]\| \leq L\left(\frac{v}{\lambda}\Delta t+\eta\|\nabla \mathscr{L}_{\mathrm{PDD}}\|\right).$$

Step 5: The Negative Sign Convention

The negative sign appears because the PDD correction reduces the error. The bound is actually:

$$\|\hat{h}[t+1]-h[t+1]\| \leq L\frac{v}{\lambda}\Delta t-\eta L\|\nabla \mathscr{L}_{\mathrm{PDD}}\|+\text{higher order terms}.$$

This is a worst-case bound; in practice, the PDD correction reduces the mobility-induced error.

Interpretation of Theorem 2:

The theorem decomposes tracking error into:

1. **Kinematic term** $L\frac{v}{\lambda}\Delta t$: represents the error from mobility-induced channel variation,
2. **Corrective term** $\eta L\|\nabla \mathscr{L}_{\mathrm{PDD}}\|$: represents the error reduction from PDD feedback.

The ratio of these terms determines the effective tracking performance:

• At low speed ($v \ll \lambda/(\Delta t)$), the PDD correction dominates,

• At high speed ($v \gg \lambda/(\Delta t)$), mobility dominates.

Generalization to Non-Rayleigh Fading: The coherence length L can be generalized to $L=\lambda/(2\pi\sqrt{1-|\rho_c|^2})$, where ρ_c is the correlation coefficient of the fading process.

Bridge to Simulation Results: This theorem explains:

• The HOF vs. speed performance in Figure 10: the kinematic term increases with speed,

• The PDD correction effect in Figure 16: PDD reduces the effective error,

• The MSE vs. velocity in Figure 15: Deep-SIC maintains stable error while DnCNN degrades,

• The mobility bound in Figure 14: physical limits on tracking performance.

7.4 *BER superiority and NRMSE convergence*

Lemma 1 (Link Reliability Improvement) Under imperfect SIC, Deep-SIC yields an exponential improvement in BER relative to ChannelInformer:

$$\frac{\mathrm{BER}_{\mathrm{ChannelInformer}}}{\mathrm{BER}_{\mathrm{Deep-SIC}}} \propto \exp\left(-\frac{\|\mathbf{h}_{\mathrm{PDD}}\|^2-\|\mathbf{h}_{\mathrm{CI}}\|^2}{N_0}\right). \tag{19}$$

Complete Derivation:

Step 1: BER Model for NOMA

For user k with channel estimate $\hat{h}_k$, the BER under QPSK modulation is:

$$\mathrm{BER}(\hat{h}_k)=\frac{1}{2}erfc\left(\sqrt{\frac{|\hat{h}_k|^2}{N_0+\sigma_{\mathrm{res}}^2}}\right),$$

where σ_{res}^2 is the residual interference variance after SIC.

Step 2: SIC Residual Analysis

The SIC residual for user k is:

$$y_{k,\text{SIC}} = h_k^H \sqrt{P_k} w_k s_k + \sum_{i=1}^{k-1} h_k^H \sqrt{P_i} w_i \tilde{s}_i + n_k,$$

where $\tilde{s}_i$ are the SIC residuals. The residual variance is:

$$\sigma_{\text{res}}^2 = \sum_{i=1}^{k-1} P_i \| h_k^H w_i \|^2 \cdot \text{BER}_i.$$

Step 3: PDD Correction Effect
Deep-SIC uses PDD to correct the channel estimate:

$$h_{\text{PDD}} = h_{\text{MMSE}} + \Delta h_{\text{PDD}},$$

where Δh_{PDD} is the PDD correction. The correction reduces the residual interference:

$$\sigma_{\text{res,PDD}}^2 = \sigma_{\text{res,CI}}^2 \cdot \exp\left(- \frac{\| \Delta h_{\text{PDD}} \|^2}{N_0} \right).$$

Step 4: BER Comparison
The ratio of BERs is:

$$\frac{\text{BER}_{\text{CI}}}{\text{BER}_{\text{PDD}}} = \frac{erfc\left(\sqrt{\frac{\| h_{\text{CI}} \|^2}{N_0 + \sigma_{\text{res,CI}}^2}} \right)}{erfc\left(\sqrt{\frac{\| h_{\text{PDD}} \|^2}{N_0 + \sigma_{\text{res,PDD}}^2}} \right)}.$$

Using the approximation $erfc(x) \approx \frac{1}{\sqrt{\pi} x} e^{-x^2}$ for large x:

$$\frac{\text{BER}_{\text{CI}}}{\text{BER}_{\text{PDD}}} \approx \sqrt{\frac{N_0 + \sigma_{\text{res,PDD}}^2}{N_0 + \sigma_{\text{res,CI}}^2}} \cdot \frac{\| h_{\text{PDD}} \|}{\| h_{\text{CI}} \|} \cdot \exp\left(\frac{\| h_{\text{PDD}} \|^2}{N_0 + \sigma_{\text{res,PDD}}^2} - \frac{\| h_{\text{CI}} \|^2}{N_0 + \sigma_{\text{res,CI}}^2} \right).$$

Simplifying and neglecting the sub-exponential factors:

$$\frac{\text{BER}_{\text{CI}}}{\text{BER}_{\text{PDD}}} \propto \exp\left(- \frac{\| h_{\text{PDD}} \|^2 - \| h_{\text{CI}} \|^2}{N_0} \right).$$

Step 5: The Negative Sign Convention
The negative sign in the exponent indicates that as the PDD-enhanced channel estimate becomes more accurate ($\| h_{\text{PDD}} \|^2 > \| h_{\text{CI}} \|^2$), the BER ratio increases (BER improves).

Bridge to Simulation Results: This lemma explains:

• The BER improvement in Figure 12: exponential reduction in BER,

• The SNR gain of >3 dB: improved effective SINR from better channel estimates,

• The PDD gain in Figure 13: 1.8× improvement in learning rate at $\varepsilon = 0.2$.

Theorem 3 (Prediction Stability) The Normalized Root Mean Square Error (NRMSE) of the Deep-SIC predictor is bounded by:

$$\text{NRMSE} \leq C\sqrt{\frac{1}{T}+\frac{1}{S}},$$

where T is the input sequence length and S is the training sample size.

Complete Derivation:

Step 1: Bias-Variance Decomposition

The prediction error decomposes into bias (from finite sequence length) and variance (from finite training samples):

$$\mathbb{E}[\|\hat{f}-f\|^2] = \mathbb{E}[\|\mathbb{E}[\hat{f}]-f\|^2] + \mathbb{E}[\|\hat{f}-\mathbb{E}[\hat{f}]\|^2].$$

Step 2: Bias Term

The Transformer's self-attention mechanism with sequence length T approximates the optimal prediction function with error:

$$\text{Bias}^2 \leq \frac{C_b}{T}.$$

This follows from the universal approximation properties of Transformers: the approximation error decays as $1/T$ for sequences of length T.

Step 3: Variance Term

The variance from finite training samples S is:

$$\text{Variance} \leq \frac{C_v}{S}.$$

This follows from standard learning theory: the estimation error of S training samples is $\mathcal{O}(1/\sqrt{S})$ for the MSE, giving variance $\mathcal{O}(1/S)$.

Step 4: Combining Terms

$$\text{NRMSE} = \frac{\sqrt{\mathbb{E}[\|\hat{f}-f\|^2]}}{\sigma_f} \leq \frac{\sqrt{C_b/T + C_v/S}}{\sigma_f}.$$

Let $C = \max(C_b, C_v)/\sigma_f$:

$$\text{NRMSE} \leq C\sqrt{\frac{1}{T}+\frac{1}{S}}.$$

Interpretation of Constants:

The constants C_b, C_v, and the combined constant C are defined in Table 3, which provides their theoretical bounds and dependencies on model parameters.

Table 3. Constants for the prediction stability bound defined in Theorem 3

Constant	Definition	Bounds
C_b	Bias constant	$\propto D_{\text{model}}^2$
C_v	Variance constant	$\propto Tr(\Sigma)$
C	Combined constant	$\leq \mathcal{O}(D_{\text{model}})$

Bridge to Simulation Results: This theorem explains:

• The convergence behavior in Figure 8: increasing S reduces variance,

• The NRMSE floor in Figure 9: the $1/T$ term limits accuracy,

• The transfer learning advantage in Algorithm 2: pre-training on large S reduces the $1/S$ term.

7.5 *Computational complexity*

Proposition 2 (Time Complexity) The time complexity per training epoch for the Transformer refinement step in Deep-SIC is:

$$\mathcal{O}(S \cdot (T^2 D_{\text{model}} + T D_{\text{model}} D_{\text{ff}} + D_{\text{model}} D_{\text{out}})). \tag{20}$$

Complete Derivation:

Step 1: Self-Attention Complexity

The multi-head self-attention with h heads, sequence length T, and embedding dimension D_{model} requires:

• Query, key, value projections: $\mathcal{O}(T D_{\text{model}}^2)$,

• Attention scores: $\mathcal{O}(T^2 D_{\text{model}}/h)$ per head,

• Output projection: $\mathcal{O}(T D_{\text{model}}^2)$.

Total attention complexity: $\mathcal{O}(T^2 D_{\text{model}} + T D_{\text{model}}^2)$.

Step 2: Feed-Forward Complexity

The feed-forward network with hidden dimension D_{ff}:

• First linear layer: $\mathcal{O}(T D_{\text{model}} D_{\text{ff}})$,

• Second linear layer: $\mathcal{O}(T D_{\text{model}} D_{\text{ff}})$.

Total feed-forward complexity: $\mathcal{O}(T D_{\text{model}} D_{\text{ff}})$.

Step 3: Output Projection

The output projection to dimension D_{out}:

$$\mathcal{O}(T D_{\text{model}} D_{\text{out}}).$$

Step 4: Total Complexity

Combining and multiplying by batch size S:

$$\mathcal{O}(S \cdot (T^2 D_{\text{model}} + T D_{\text{model}}^2 + T D_{\text{model}} D_{\text{ff}} + T D_{\text{model}} D_{\text{out}})).$$

Simplifying: $\mathcal{O}(S \cdot (T^2 D_{\text{model}} + T D_{\text{model}} D_{\text{ff}} + D_{\text{model}} D_{\text{out}}))$ since $D_{\text{model}} > T$ typically.

Step 5: Per-User Complexity

The full Deep-SIC pipeline processes users independently:

$$\mathcal{O}(K \cdot \text{Transformer complexity}) = \mathcal{O}(K).$$

The SIC process is sequential but uses the same Transformer per user.

Bridge to Simulation Results: This proposition explains:

• The linear scaling in Figure 11: $\mathcal{O}(K)$ complexity,

• The FLOPs advantage over pure Transformer: $\mathcal{O}(K)$ vs $\mathcal{O}(K^2)$,

• The practical feasibility for dense 5G deployments.

8. Conclusion and future works

This paper introduced the Deep-SIC framework, which transforms channel estimation from a reactive process into a predictive, self-correcting mechanism for intelligent handover management. It utilises a Transformer architecture to learn from long-term temporal dependencies and uses PDD as a feedback signal. Simulations at the system level demonstrate that this approach decreases the HOF rate and the ping-pong effect and improves QoS for cell-edge users. The framework scales linearly with the number of users, reducing computational cost. Additionally, the totality of theoretical assurances, algorithm architecture, and system-level validation demonstrates that Deep-SIC is foreseeable, effective, and adaptable, enhancing handover performance and user experience in an actual 5G NOMA implementation.

Future research will incorporate validation using real-world channel traces and hardware-in-the-loop testing. The Deep-SIC paradigm will be extended to broader networking contexts. This extension includes evaluating robustness under heterogeneous traffic conditions, particularly mixed URLLC (Ultra-Reliable Low-Latency Communication) and mMTC (massive Machine-Type Communication) traffic, to ensure applicability across diverse 5G and 6G scenarios. Additionally, we aim to extend the framework to predict other network-level key performance indicators (KPIs), such as anticipated load or interference patterns, to facilitate joint optimisation of handover, resource allocation, and network slicing decisions. Finally, we will investigate integration with Open RAN (O-RAN) architectures, focusing on how the Deep-SIC predictor can serve as a near-real-time RIC (RAN Intelligent Controller) application for intelligent radio resource management.

Abbreviation

NOMA	Non-Orthogonal Multiple Access
SIC	Successive Interference Cancellation
CSI	Channel State Information
PDD	Partially Decoded Data
HOF	Handover Failure
RSRQ	Reference Signal Receive Quality
MMSE	Minimum Mean Square Error
SINR	Signal-to-Interference-plus-Noise Ratio
NOMA-HO	NOMA Handover
UE	User Equipment
BS	Base Station
DnCNN	Denoising Convolutional Neural Network
SINR	Signal-to-Interference-plus-Noise Ratio

NRMSE	Normalized Root Mean Squared Error
MASE	Mean Absolute Scaled Error
MSE	Mean Squared Error
R^2	R-squared
BER	Bit Error Rate

Author Contributions

Conceptualization, S.M. and P.M.; methodology, S.M.; software, S.M.; validation, S.M., K.S. and P.M.; formal analysis, S.M.; investigation, S.M.; resources, P.M.; data curation, S.M.; writing-original draft preparation, S.M.; writing-review and editing, S.M. and K.S.; visualization, S.M.; supervision, P.M.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data Availability Statement

The augmented dataset used in this study is derived from publicly available sources and will be released in a public repository. The base dataset is available online at [26], and the augmented version will be made available at [27] upon publication.

Acknowledgements

The authors express their thanks to the people helping with this work. In the preparation of this work, the authors used Gemini to assist with text editing and formatting refinement. After using this tool/service, the authors reviewed and edited the content as needed, and take full responsibility for the content of the publication.

Conflicts of Interest

The authors declare no competing financial interest.

References

[1] G. Bacci, A. A. D'Amico, and L. Sanguinetti, "Mmse channel estimation in large-scale mimo: Improved robustness with reduced complexity," *IEEE Transactions on Wireless Communications*, vol. 23, no, 12, pp. 18563–18575, 2024.

[2] H. Soury, B. Smida, and S. M. Aliabadi, "Accurate mmse expressions for short-packet pilot-based channel estimation," *IEEE Wireless Communications Letters*, vol. 12, no. 12, pp. 2188–2192, 2023.

[3] M. Meenalakshmi, S. Chaturvedi, and V. K. Dwivedi, "Enhancing channel estimation accuracy in polar-coded mimo–ofdm systems via cnn with 5g channel models," *AEU-International Journal of Electronics and Communications*, vol. 173, p. 155016, 2024.

[4] H. Chen, N. Li, R. Long, and Y. C. Liang, "Channel estimation and training design for active ris aided wireless communications," *IEEE Wireless Communications Letters*, vol. 12, no. 11, pp. 1876–1880, 2023.

[5] P. R. Gomes, G. T. de Araújo, B. Sokal, A. L. de Almeida, B. Makki, and G. Fodor, "Channel estimation in ris-assisted mimo systems operating under imperfections," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14200–14213, 2023.

[6] X. Gong, W. Chen, L. Sun, J. Chen, and B. Ai, "An esprit-based supervised channel estimation method using tensor train decomposition for mmwave 3-d mimo-ofdm systems," *IEEE Transactions on Signal Processing*, vol. 71, pp. 555–570, 2023.

[7] A. Lawal, Q. Mayyala, K. Abed-Meraim, N. Iqbal, and A. Zerguine, "Blind signal estimation using structured subspace technique," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 3007–3011, 2021.

[8] V. Vasylyshyn, "Channel estimation method for ofdm communication system using adaptive singular spectrum analysis," In *Proc. IEEE International Conference on Electronics and Nanotechnology (ELNANO)*, 2020, pp. 884–887.

[9] T. Jiang, F. Xie, S. Yuan, S. Yao, K. Wu, and X. Wang, "Implementation of matrix svd decomposition module for subspace channel estimation," In *Proc. IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019, pp. 1096–1102.

[10] I. Khatri, T. Acharya, A. Annamalai, and M. Chouikha, "Higher order statistics of channel capacity in $\kappa - \mu$ fading channel," In *Proc. International Conference on Ubiquitous and Future Networks (ICUFN)*, 2021, pp. 35–40.

[11] S. Kharbech, E. P. Simon, A. Belazi, and W. Xiang, "Denoising higher-order moments for blind digital modulation identification in multiple-antenna systems," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 765–769, 2020.

[12] J. Liu, X. L. Huang, F. Hu, and S. Yu, "Higher-order cumulant-assisted constant wideband compressive spectrum sensing using semantic correlation mining," *IEEE Transactions on Wireless Communications*, vol. 24, no. 2, pp. 893–908, 2024.

[13] A. S. Alwakeel and A. H. Mehana, "Data-aided channel estimation for multiple-antenna users in massive mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10752–10760, 2019.

[14] Y. S. Jeon, J. Li, N. Tavangaran, and H. V. Poor, "Data-aided channel estimator for mimo systems via reinforcement learning," In *Proc. IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[15] I. Khan, M. Cheffena, and M. M. Hasan, "Data aided channel estimation for mimo-ofdm wireless systems using reliable carriers," *IEEE Access*, vol. 11, pp. 47836–47847, 2023.

[16] P. Sabeti, A. Farhang, I. Macaluso, N. Marchetti, and L. Doyle, "Blind channel estimation for massive mimo: A deep learning assisted approach," In *Proc. IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[17] R. Crepaldi, J. Lee, R. Etkin, S. Lee, and R. Kravets, "CSI-SF: Estimating wireless channel state using CSI sampling & fusion," In *Proc. IEEE INFOCOM*, 2012, pp. 154–162.

[18] A. Al Hammadi, P. C. Sofotasios, S. Muhaidat, M. Al-Qutayri, and H. Elgala, "Non-orthogonal multiple access for hybrid vlc-rf networks with imperfect channel state information," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 398–411, 2020.

[19] H. H. Choi, H. Kim, J. Na, and H. Lee, "Soft non-orthogonal multiple access-based handover for throughput enhancement," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 10197–10202, 2022.

[20] Y. Chen, M. Wen, L. Wang, W. Liu, and L. Hanzo, "Sinr-outage minimization of robust beamforming for the non-orthogonal wireless downlink," *IEEE Transactions on Communications*, vol. 68, pp. 7247–7257, 2020.

[21] I. Hur and S. Lee, "Handover and radio link performance analysis on standalone non-orthogonal multiple access (noma) environment," In *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, 2024, pp. 1–5.

[22] A. Çalhan and M. Cicioğlu, "Handover scheme for 5g small cell networks with non-orthogonal multiple access," *Computer Networks*, vol. 183, p. 107601, 2020.

[23] S. Abhirami, N. S. Mouni, S. D. Amuru, and A. Kumar, "Handover algorithms for enhanced throughput in a hybrid oma-noma system with imperfect sic," In *Proc. IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, 2024, pp. 1–5.

[24] A. S. Marcano and H. L. Christiansen, "Impact of noma on network capacity dimensioning for 5g hetnets," *IEEE Access*, vol. 6, pp. 13587–13603, 2018.

[25] Y. Liu, S. Zhang, X. Mu, Z. Ding, R. Schober, N. Al-Dhahir, et al., "Evolution of noma toward next generation multiple access (ngma) for 6g," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 4, pp. 1037–1071, 2022.

[26] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," In *Proc. ACM Multimedia Systems Conf. (MMSys)*, 2020, pp. 303–308.

[27] S. Majhi, "Deep-SIC Channel Estimation Data and Code Repository," *GitHub*, 2026. [Online]. Available: https://github.com/sumitamajhi/Deep-SIC-Channel-Estimation-Data-and-Code-Repository/tree/main, [Accessed Jun. 10, 2026].

[28] S. Pateriya, S. Bandopadhaya, A. K. Bairwa, and V. Jain, "Optimizing 5g nr link layer parameters for embb and urllc applications under dynamic channel and transmission configurations," *Scientific Reports*, vol. 16, 2026.

[29] A. B. Belguidoum, M. L. Tounsi, and S. Mekaoui, "Optimization of 5g throughput and latency in non-standalone and standalone mode," *Telecommunication Systems*, vol. 88, p. 30, 2025.

[30] A. Kumar, M. K. Sharma, K. Sengar, and S. Kumar, "Noma based cr for qam-64 and qam-256," *Egyptian Informatics Journal*, vol. 21, no. 2, pp. 67–71, 2020.

[31] E. T. Kekeç, M. S. Gençağ, and K. Boyali, "Analysis of lte network connection parameters at legal maximum uav flight altitudes and comparison with ground," *Journal of Aviation*, vol. 9, no. 3, pp. 508–515, 2025.

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

Appendix. Proof of Proposition 1

Theorem 4 (Convergence of PDD-Corrected Transformer Updates) Under the following assumptions:

1. The channel estimation loss $\mathscr{L}(\mathbf{h}) = \|\mathbf{y} - \mathbf{H}(\mathbf{h})\mathbf{x}\|^2$ is γ-smooth with Lipschitz constant L_γ.
2. The PDD gradient correction term $\psi(\mathbf{PDD})$ is β-Lipschitz continuous.
3. The Jacobian $\mathbf{J} = \nabla_{\mathbf{h}}(\mathbf{H}(\mathbf{h})\mathbf{x})$ is bounded such that $\|\mathbf{J}\|_2 \leq J_{\max}$.
4. The error in the partially decoded data (PDD) is bounded by $\varepsilon_{\text{PDD}} < \frac{\eta\gamma}{2\|\mathbf{W}_{\text{att}}\|_F}$, where γ is the effective gradient margin (smallest eigenvalue of $\nabla^2\mathscr{L}$), and $\mathbf{W}_{\text{att}}$ is the weight matrix of the self-attention layer.

The update rule:

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} - \eta[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})]$$

converges linearly to the optimal channel estimate $\mathbf{h}^*$ when the learning rate satisfies:

$$0 < \eta < \frac{2}{\lambda_{\max}(\mathbf{J}^T\mathbf{J}) + \beta},$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue.

Proof:

Step 1: Reformulate the update rule.

The channel estimate update using the PDD is given by:

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} - \eta[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})] = \mathbf{h}^{(t)} - \eta[\mathbf{J}^T(\mathbf{y} - \mathbf{H}(\mathbf{h}^{(t)})\mathbf{x}) + \psi(\mathbf{PDD}^{(t)})].$$

Step 2: Analyze the error dynamics.

Let $\mathbf{e}^{(t)} = \mathbf{h}^{(t)} - \mathbf{h}^*$ be the estimation error at iteration t. Then:

$$\mathbf{e}^{(t+1)} = \mathbf{h}^{(t)} - \eta[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})] - \mathbf{h}^* = \mathbf{e}^{(t)} - \eta[\nabla\mathscr{L}(\mathbf{h}^{(t)}) + \psi(\mathbf{PDD}^{(t)})].$$

Taking norms and applying the triangle inequality:

$$\|\mathbf{e}^{(t+1)}\| \leq \|\mathbf{e}^{(t)} - \eta\nabla\mathscr{L}(\mathbf{h}^{(t)})\| + \eta\|\psi(\mathbf{PDD}^{(t)})\|.$$

Step 3: Bound the loss gradient term.

Using the γ-smoothness of $\mathscr{L}$ (Assumption 1) and the mean value theorem:

$$\|\mathbf{e}^{(t)} - \eta\nabla\mathscr{L}(\mathbf{h}^{(t)})\| \leq \|\mathbf{I} - \eta\nabla^2\mathscr{L}(\mathbf{h}^{(t)})\| \cdot \|\mathbf{e}^{(t)}\|.$$

Since $\nabla^2\mathscr{L}(\mathbf{h}^{(t)}) \approx \mathbf{J}^T\mathbf{J}$ near the optimum, we have:

$$\|\mathbf{I} - \eta \nabla^2 \mathscr{L}(\mathbf{h}^{(t)})\| \leq \rho(\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}),$$

where $\rho(\cdot)$ denotes the spectral radius.

Step 4: Bound the PDD correction term.

Using the β-Lipschitz continuity of ψ (Assumption 2):

$$\|\psi(\mathbf{PDD}^{(t)})\| \leq \beta \|\mathbf{e}^{(t)}\| + \alpha \varepsilon_{\text{PDD}},$$

where α is a constant relating PDD error to gradient perturbation. Specifically, $\alpha = \|\mathbf{W}_{\text{att}}\|_F$, which bounds how the PDD error propagates through the self-attention mechanism.

Step 5: Combine error terms.

Substituting into the error bound:

$$\|\mathbf{e}^{(t+1)}\| \leq \rho(\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}) \cdot \|\mathbf{e}^{(t)}\| + \eta \beta \|\mathbf{e}^{(t)}\| + \eta \alpha \varepsilon_{\text{PDD}} = (\rho(\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}) + \eta \beta) \|\mathbf{e}^{(t)}\| + \eta \alpha \varepsilon_{\text{PDD}}.$$

Step 6: Enforce contraction mapping.

For linear convergence, we require:

$$\rho(\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}) + \eta \beta < 1.$$

Using the eigenvalue bound $\lambda_{\max}(\mathbf{J}^T \mathbf{J}) \leq J_{\max}^2$:

$$\max_i |1 - \eta \lambda_i(\mathbf{J}^T \mathbf{J})| + \eta \beta < 1.$$

This yields the learning rate condition:

$$\eta < \frac{2}{\lambda_{\max}(\mathbf{J}^T \mathbf{J}) + \beta}.$$

Step 7: Control PDD error perturbation.

From the PDD error bound (Assumption 4):

$$\varepsilon_{\text{PDD}} < \frac{\eta \gamma}{2\|\mathbf{W}_{\text{att}}\|_F}.$$

This ensures that the perturbation term $\eta \alpha \varepsilon_{\text{PDD}}$ remains dominated by the contraction term when:

$$\eta \alpha \varepsilon_{\text{PDD}} \ll (\rho(\mathbf{I} - \eta \mathbf{J}^T \mathbf{J}) + \eta \beta) \|\mathbf{e}^{(t)}\|.$$

Substituting $\alpha = \|\mathbf{W}_{\text{att}}\|_F$ and the PDD error bound:

$$\eta\|\mathbf{W}_{\text{att}}\|_F \cdot \frac{\eta\gamma}{2\|\mathbf{W}_{\text{att}}\|_F} = \frac{\eta^2\gamma}{2} \ll (\rho(\mathbf{I} - \eta\mathbf{J}^T\mathbf{J}) + \eta\beta)\|\mathbf{e}^{(t)}\|.$$

This holds for sufficiently small η, which is ensured by the learning rate condition.

Step 8: Verify convergence.

Under both conditions, the error decreases geometrically:

$$\|\mathbf{e}^{(t+1)}\| \leq (\rho(\mathbf{I} - \eta\mathbf{J}^T\mathbf{J}) + \eta\beta)\|\mathbf{e}^{(t)}\| + \eta\alpha\varepsilon_{\text{PDD}} < \|\mathbf{e}^{(t)}\|$$

for $\|\mathbf{e}^{(t)}\| > \frac{\eta\alpha\varepsilon_{\text{PDD}}}{1-\rho(\mathbf{I}-\eta\mathbf{J}^T\mathbf{J})-\eta\beta}$.

Thus, the error converges to a bounded neighbourhood of zero, with the bound proportional to ε_{PDD}, confirming the claimed exponential convergence with a steady-state error floor.

Step 9: Connection to Algorithm 1.

Algorithm 1 implements this update through the MMSE initial estimate $\mathbf{h}^{(0)}$, SIC process generating $\mathbf{PDD}^{(t)}$, Transformer's self-attention computing $\psi(\mathbf{PDD}^{(t)})$ through weighted combination, and the feed-forward layers applying the gradient update. The convergence conditions in Proposition 1 ensure Algorithm 1 converges to the optimal channel estimate despite imperfect SIC.