Universidade Estadual de Londrina

RAFAEL MARASCA MARTINS

INTELLIGENT PHYSICAL LAYER ARCHITECTURES FOR 6G

LONDRINA
2026

RAFAEL MARASCA MARTINS

INTELLIGENT PHYSICAL LAYER ARCHITECTURES FOR 6G

Dissertação apresentada ao Programa de Mestrado em Engenharia Elétrica da Universidade Estadual de Londrina - UEL, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Taufik Abrão

Coorientador: Prof. Dr. Luis Carlos Mathias

LONDRINA
2026

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Martins, Rafael Marasca.
Intelligent Physical Layer Architectures for 6G / Rafael Marasca Martins. - Londrina, 2026.
161 f.

Orientador: Taufik Abrão.
Coorientador: Luis Carlos Mathias.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Estadual de Londrina, Centro de Tecnologia e Urbanismo, Programa de Pós-Graduação em Engenharia Elétrica, 2026.
Inclui bibliografia.

1. Telecomunicações - Tese. 2. 6G - Tese. 3. Aprendizado de Máquina - Tese. 4. Camada Física - Tese. I. Abrão, Taufik. II. Mathias, Luis Carlos. III. Universidade Estadual de Londrina. Centro de Tecnologia e Urbanismo. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

CDU 62

RAFAEL MARASCA MARTINS

INTELLIGENT PHYSICAL LAYER ARCHITECTURES FOR 6G

Dissertação apresentada ao Programa de Mestrado em Engenharia Elétrica da Universidade Estadual de Londrina - UEL, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Taufik Abrão
Coorientador: Prof. Dr. Luis Carlos Mathias

BANCA EXAMINADORA

Prof. Dr. Glauber Gomes de Oliveira Brante
Universidade Tecnológica Federal do Paraná – UTFPR

Prof. Dr. Richard Demo Souza
Universidade Federal de Santa Catarina – UFSC

Prof. Dr. Taufik Abrão
Universidade Estadual de Londrina

Londrina, 19 de Junho de 2026.

ACKNOWLEDGEMENTS

To my family, especially my parents, for their unconditional support, for their patience during my moments of absence, and for believing in my dreams even when the path seemed difficult. Without your emotional support, none of this would be possible.

To my advisor, Prof. Dr. Taufik Abrão, for his precise guidance, for the trust placed in my work, and for sharing his vast knowledge, challenging me to always strive for excellence.

To my co-advisor, Prof. Dr. Luis Carlos Mathias, for his valuable contributions, his constant availability, and his attentive eye that helped give final shape to this project and to all the work developed during my master's degree.

MARTINS, R. M.. **Arquiteturas Inteligentes da Camada Física para 6G**. 2026. 160f. – Universidade Estadual de Londrina, Londrina, 2026.

RESUMO

A camada física de redes sem fio de próxima geração enfrenta desafios que vão além dos limites teóricos clássicos, sendo frequentemente limitada por restrições práticas, como orçamentos computacionais restritos e não idealidades de hardware. Esta dissertação explora como a incorporação de inteligência na camada física pode superar esses gargalos em três cenários relevantes para redes 6G. Primeiramente, no contexto de comunicações assistidas por superfícies inteligentes reconfiguráveis (RIS), o descasamento entre a baixa taxa de atualização do posicionamento GNSS e a necessidade de reconfiguração rápida da superfície compromete o alinhamento do feixe. Para contornar essa limitação, uma estratégia de controle preditivo baseada no filtro de Kalman é proposta, empregando modelos de velocidade constante e aceleração constante. A abordagem preditiva reduz a degradação da potência recebida em até aproximadamente cinco vezes em relação a uma estratégia reativa convencional, com o modelo de aceleração constante se mostrando mais robusto para trajetórias com curvas e mudanças abruptas de direção. O segundo cenário de aplicação desenvolve-se no domínio de sistemas de comunicação e sensoriamento integrados (ISAC): a otimização conjunta da conformação de feixe para sensoriamento e comunicação resulta em problemas não convexos cuja solução iterativa convencional frequentemente falha em convergir dentro de tempos computacionais realistas. Para superar esse gargalo, uma arquitetura de desdobramento profundo é desenvolvida, reformulando o algoritmo de aproximação convexa sucessiva como um grafo computacional com parâmetros treináveis. O otimizador resultante eleva a taxa de factibilidade de SINR sob o mesmo número de iterações, além de proporcionar ganhos na capacidade de sensoriamento. A terceira aplicação auxiliada por inteligência na camada física refere-se às comunicações ópticas sem fio: a utilização de componentes ópticos comerciais na modulação por troca de cores (CSK) do padrão IEEE 802.15.7 introduz distorções na constelação recebida, o que torna o detector geométrico convencional inadequado. Um modelo de sistema ponta a ponta é desenvolvido e três detectores baseados em aprendizado de máquina são avaliados ec omparados com o detector tradicional, demonstrando que a distorção geométrica da constelação, e não o ruído aditivo, constitui o principal limitante de desempenho. Os detectores baseados em aprendizado de máquina reduzem o erro de classificação em aproximadamente 51%. Em todos os cenários investigados, a inteligência incorporada à camada física, seja por meio de predição baseada em modelo, desdobramento de algoritmos ou classificação orientada por dados, proporciona ganhos expressivos e praticamente realizáveis, evidenciando o potencial dessas técnicas para as futuras redes 6G.

Palavras-chave: Aprendizado de máquina; 6G; Superfícies Refletoras Inteligentes; Sensoriamento e Comunicação Integrados; Comunicação por luz visível; Camada física; Esquemas de transmissão.

MARTINS, R. M.. **Intelligent Physical Layer Architectures for 6G**. 2026. 160p. – State University of Londrina, Londrina, 2026.

ABSTRACT

The physical layer of next-generation wireless networks faces challenges that extend beyond classical theoretical limits and are frequently constrained by practical issues, such as restricted computational budgets and hardware non-idealities. This dissertation explores how incorporating intelligence into the physical layer can overcome these bottlenecks across three scenarios relevant to 6G networks. First, in Reconfigurable Intelligent Surface (RIS)-aided communications, the mismatch between the low update rate of GNSS positioning and the need for rapid surface reconfiguration compromises beam alignment. To address this limitation, a predictive control strategy based on the Kalman filter is proposed, employing constant-velocity and constant-acceleration motion models. This predictive approach reduces received power degradation by a factor of up to five compared to a conventional reactive strategy, with the constant-acceleration model proving more robust for trajectories involving curves and abrupt direction changes. The second application scenario occurs in Integrated Sensing and Communication (ISAC) systems, where joint beamforming optimization for sensing and communication leads to non-convex problems whose conventional iterative solutions often fail to converge within realistic computational budgets. To overcome this bottleneck, a deep unfolding architecture is developed that reformulates the Successive Convex Approximation algorithm as a computational graph with trainable parameters. The resulting optimizer increases the feasibility rate under the same iteration count while also providing gains in sensing capabilities. The third intelligence-aided application at the physical layer pertains to optical wireless communications: the use of off-the-shelf optical components in IEEE 802.15.7 Color Shift Keying (CSK) modulation introduces distortions in the received constellation that render the conventional geometric detector inadequate. An end-to-end system model is developed, and three machine-learning-based detectors are evaluated, demonstrating that geometric constellation warping, rather than additive noise, constitutes the dominant performance limiter. The proposed detectors reduce the classification error by approximately 51%. Across all investigated scenarios, the intelligence incorporated into the physical layer—whether through model-based prediction, algorithm unfolding, or data-driven classification—yields substantial and practically realizable gains, highlighting the potential of these techniques for future 6G networks.

Keywords: Machine learning; 6G; Intelligent reflecting surfaces; Integrated Sensing and Communication; Visible light communication; Physical layer; Transmission schemes.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Adam Adaptive Moment Estimation.

ADC Analog-to-Digital Converter.

ADMM Alternating Direction Method of Multipliers.

AUC Area Under Curve.

AWGN Additive White Gaussian Noise.

B5G Beyond 5G.

BS Base Station.

C.I. Confidence Interval.

CA Constant-Acceleration.

CDF Cumulative Distribution Function.

CNN Convolutional Neural Network.

CRB Cramér-Rao Bound.

CSI Channel State Information.

CSK Color Shift Keying.

CV Constant-Velocity.

DU Deep Unfolding.

DU-SCA Deep Unfolded SCA.

EMBB Enhanced Mobile Broadband.

FD Full-Duplex.

FIM Fisher Information Matrix.

FNN Feedforward Neural Network.

FPGA Field-Programmable Gate Array.

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

HBF Hybrid Beamforming.

IMM-KF Interacting Multiple Model Kalman Filter.

IRS Intelligent Reflecting Surface.

ISAC Integrated Sensing and Communication.

JCAS Joint Communications and Sensing.

JRC Joint Radar-Communication.

KAN Kolmogorov-Arnold Network.

KKT Karush-Kuhn-Tucker.

kNN k-Nearest Neighbors.

LED Light-Emitting Diode.

LoS Line-of-Sight.

MC-MIMO Multicarrier Multiple-Input Multiple-Output.

MED Minimum Euclidean Distance.

MIMO Multiple-Input Multiple-Output.

MISO Multiple-Input Single-Output.

MLE Maximum Likelihood Estimator.

MLP Multilayer Perceptron.

MMTC Massive Machine-Type Communications.

MSE Mean Squared Error.

MU Multi-User.

MU-MIMO Multi-User Multiple-Input Multiple-Output.

MUI Multi-User Interference.

NAGD Nesterov Aceelerated Gradient Descent.

NLoS Non-Line-of-Sight.

OF Optical Filter.

OTS Off the Shelf.

PCA Principal Component Analysis.

PGA Projected Gradient Descent.

QoS Quality of Service.

ReLU Rectified Linear Unit.

RF Radio Frequency.

RIS Reconfigurable Intelligent Surface.

RMO Riemannian Manifold Optimization.

ROC Receiver Operating Characteristic.

SCA Successive Convex Approximation.

SDR Semidefinite Relaxation.

SINR Signal to Interference plus Noise Ratio.

SNR Signal-to-Noise Ratio.

SOC Second-Order Cone.

SOCP Second-Order Cone Program.

SSCA Stochastic Successive Convex Approximation.

TIA Transimpedance Amplifier.

UE User Equipment.

URLLC Ultra Reliable Low-Latency Communication.

VLC Visible Light Communication.

CONTENTS

1 INTRODUCTION

1.1 Motivation

The evolution toward sixth-generation (6G) wireless networks is expected to bring transformative changes to the design and operation of communication systems (Hakeem; Hussein; Kim, 2022). Whereas previous generations focused primarily on increasing data rates and expanding coverage, 6G is envisioned as a platform that simultaneously supports Enhanced Mobile Broadband (EMBB), Massive Machine-Type Communications (MMTC), Ultra Reliable Low-Latency Communication (URLLC), and emerging functionalities such as integrated sensing, positioning, and environmental awareness (Hakeem; Hussein; Kim, 2022; Chen *et al.*, 2023). Realizing this vision demands a fundamental rethinking of the physical layer, which must become more adaptive, resource-efficient, and capable of operating under increasingly complex and heterogeneous propagation conditions (Singh *et al.*, 2025).

A key enabler of this transformation is the incorporation of intelligence into the physical layer itself. Traditional physical layer design relies on fixed, analytically derived processing rules, such as matched filters, maximum-likelihood detectors, and iterative optimizers, that are optimal under idealized assumptions but often exhibit significant performance degradation when confronted with practical constraints such as hardware nonidealities, limited computational budgets, sparse measurements, or model mismatches (Singh *et al.*, 2025; Saad; Bennis; Chen, 2020; Zhang *et al.*, 2019). In recent years, a growing body of research has demonstrated that learning-based and model-driven techniques can systematically address these limitations by adapting processing strategies to the specific operating conditions encountered in practice (He *et al.*, 2019; Monga; Li; Eldar, 2021).

This trend is manifest across several physical layer domains that are central to the 6G vision. In Reconfigurable Intelligent Surface (RIS)-aided communications, intelligent surfaces with hundreds or thousands of reconfigurable elements offer unprecedented control over the propagation environment, but their effective operation requires real-time phase configuration that is tightly coupled to channel conditions and user mobility (Liu *et al.*, 2021; Wu *et al.*, 2021; ElMossallamy *et al.*, 2020). In Integrated Sensing and Communication (ISAC) systems, the transmitter must jointly optimize sensing and communication performance under shared spectral and power resources, leading to non-convex optimization problems whose real-time solution is computationally prohibitive (Liu *et al.*, 2020, 2022b; Liu *et al.*, 2022a). In optical wireless communications, particularly Visible Light Communication (VLC) based on standardized Color Shift Key-

ing (CSK) modulation, practical off-the-shelf components introduce spectral overlap, nonlinear distortions, and inter-channel crosstalk that defeat conventional geometric detectors (Jovicic; Li; Richardson, 2013; Pathak *et al.*, 2015; Chowdhury *et al.*, 2018). In each of these domains, the core challenge is not a lack of theoretical understanding but rather a mismatch between the assumptions of classical algorithms and the realities of practical deployment.

This dissertation addresses three such mismatches by developing intelligent physical layer architectures that leverage prediction, algorithm unrolling, and supervised learning to bridge the gap between theoretical performance and practical operating conditions.

1.2 Objectives

1.2.1 General Objective

The general objective of this dissertation is to investigate, develop, and evaluate intelligent physical layer architectures for next-generation wireless communication systems, demonstrating that the systematic incorporation of learning-based and model-driven intelligence into established physical layer building blocks yields substantial performance gains under realistic operating conditions.

1.2.2 Specific Objectives

Three specific objectives are pursued, each addressing a distinct physical layer design challenge:

1. **Predictive RIS control under sparse positioning.** Develop a Kalman-filter-based predictive control strategy for RIS-aided communications that anticipates user equipment trajectories from low-rate Global Navigation Satellite System (GNSS) updates, enabling proactive beam steering between consecutive positioning measurements. The objective is to quantify the improvement in received-power maintenance relative to a naïve reactive baseline across different motion models, trajectory profiles, and sampling rates.

2. **Low-latency joint sensing and communication beamforming.** Propose a deep unfolding architecture that reformulates the classical Successive Convex Approximation (SCA) algorithm for ISAC beamforming as a fixed-depth computational graph with learnable hyperparameters. The objective is to demonstrate that the unfolded optimizer achieves substantially higher feasibility rates and superior sensing performance compared to the conventional solver under identical iteration budgets.

3. **Machine-learning-based detection for practical CSK.** Develop and compare data-driven detectors, namely k-Nearest Neighbors (kNN), Feedforward Neural Network (FNN), and Rectified Linear Unit (ReLU)-Kolmogorov-Arnold Network (KAN), for IEEE 802.15.7 CSK modulation with off-the-shelf optical components. The objective is to quantify the accuracy improvement over the conventional Minimum Euclidean Distance (MED) across different constellation orders and noise levels, and to establish practical deployment guidelines regarding sample efficiency, computational cost, and noise-regime suitability.

1.3 Contributions

The main contributions of this dissertation are summarized as follows:

- A predictive RIS control framework based on Kalman filtering with Constant-Velocity (CV) and Constant-Acceleration (CA) motion models is proposed and evaluated. The framework is shown to reduce received-power degradation by up to approximately fivefold compared to a naïve baseline at low GNSS update rates, and the CA model is identified as more robust for trajectories with turning behavior or abrupt acceleration changes. This contribution resulted in a paper accepted and presented at the XLIII Brazilian Symposium on Telecommunications and Signal Processing (SBrT 2025), a copy of which is provided in Appendix A.

- A model-driven deep unfolding architecture for ISAC beamforming is developed, in which the SCA algorithm is unrolled into a fixed-depth computational graph with learnable step sizes, momentum coefficients, and penalty weights. The proposed Deep Unfolding (DU)-SCA framework achieves a nearly eightfold improvement in Signal to Interference plus Noise Ratio (SINR) feasibility rate and 1.25–1.75 dB gains in Cramér-Rao Bound (CRB) relative to the classical solver under the same iteration budget. This work has been submitted for publication to the IEEE Transactions on Wireless Communications (TWC), where its manuscript is included in Appendix C.

- A comprehensive end-to-end system model for IEEE 802.15.7 CSK with practical off-the-shelf optical components is developed, and three supervised learning detectors are compared against the conventional MED. The study demonstrates that constellation warping, not additive noise, is the dominant performance limiter, and that learned detectors eliminate nearly all of the approximately 29% structural error floor present in the near-noiseless 16-CSK case, while at the operating-point noise they reduce the classification error by approximately 51%. Practical guidelines are provided regarding which detector is best suited for each noise regime

and constellation order. This study forms the basis of a manuscript submitted to the IEEE/Optica Journal of Lightwave Technology (JLT), which is attached in Appendix B.

1.4 Document Organization

The remainder of this dissertation is organized as follows.

Chapter 2 addresses the predictive control of RISs in beyond-5G networks. A Kalman-filter-based framework employing CV and CA motion models is proposed to maintain beam alignment during GNSS positioning gaps. Numerical results are presented for three trajectory profiles at multiple sampling rates.

Chapter 3 presents a deep unfolded SCA architecture for non-convex ISAC beamforming optimization. The classical iterative solver is reformulated as a differentiable computational graph with learnable hyperparameters, and the resulting optimizer is evaluated in terms of feasibility rate, CRB, and learned parameter interpretability.

Chapter 4 investigates the detection problem in IEEE 802.15.7 CSK modulation with practical off-the-shelf optical components. An end-to-end system model is developed, and four detection strategies, namely MED, kNN, FNN, and ReLU-KAN, are compared across constellation orders and noise levels.

Finally, chapter 5 presents the specific and general conclusions of the dissertation.

2 PREDICTIVE CONTROL FOR RIS-AIDED B5G NETWORKS USING KALMAN FILTERS

2.1 Background and Motivation

Overcoming multipath fading and signal blockage remains a fundamental challenge in modern wireless communications, particularly within dense urban networks (ElMossallamy *et al.*, 2020; Chen *et al.*, 2023). In these environments, signals experience severe scattering, reflection, and attenuation as they interact with macroscopic urban infrastructure (Huang *et al.*, 2020; ElMossallamy *et al.*, 2020). Traditionally, this wireless propagation channel has been treated as a highly probabilistic and uncontrollable entity, forcing communication systems to computationally adapt to impairments rather than actively alter the environment itself (ElMossallamy *et al.*, 2020).

To disrupt this paradigm, RIS has emerged as a transformative technology enabling the realization of smart, programmable radio environments (ElMossallamy *et al.*, 2020; Liu *et al.*, 2021). As illustrated in Fig. 1, an RIS is physically realized as a planar metasurface comprising a massive array of low-cost, sub-wavelength scattering elements, frequently referred to as meta-atoms (Costa; Borgese, 2021; ElMossallamy *et al.*, 2020). Governed by a smart controller, these discrete elements dynamically interact with incident electromagnetic waves.

While active RIS architectures capable of signal amplification have recently emerged, the predominant and most energy-efficient deployments operate passively (Wu; Zhang, 2020; ElMossallamy *et al.*, 2020). In these passive configurations, rather than actively transmitting or amplifying signals, the RIS functions by electronically reconfiguring the electromagnetic boundary conditions of the surface (Wu; Zhang, 2020; ElMossallamy *et al.*, 2020). By adjusting the surface impedance of these meta-atoms, typically via integrated tunable components such as PIN diodes or varactors, the controller applies a specific phase-shift matrix to the impinging wave (ElMossallamy *et al.*, 2020; Costa; Borgese, 2021). This highly granular, element-wise manipulation enables the RIS to sculpt the reflected wavefront, coherently steering the beam to induce constructive interference at the targeted User Equipment (UE). Consequently, strategic RIS deployment not only maximizes spectral efficiency but also mitigates severe shadowing by bypassing physical blockages, synthesizing a robust virtual Line-of-Sight (LoS) link between the Base Station (BS) and the UE (Liu *et al.*, 2021; Chen *et al.*, 2023).

Despite the immense potential of RIS, practical integration into wireless networks introduces several critical challenges (Wu *et al.*, 2021). Primarily, the passive

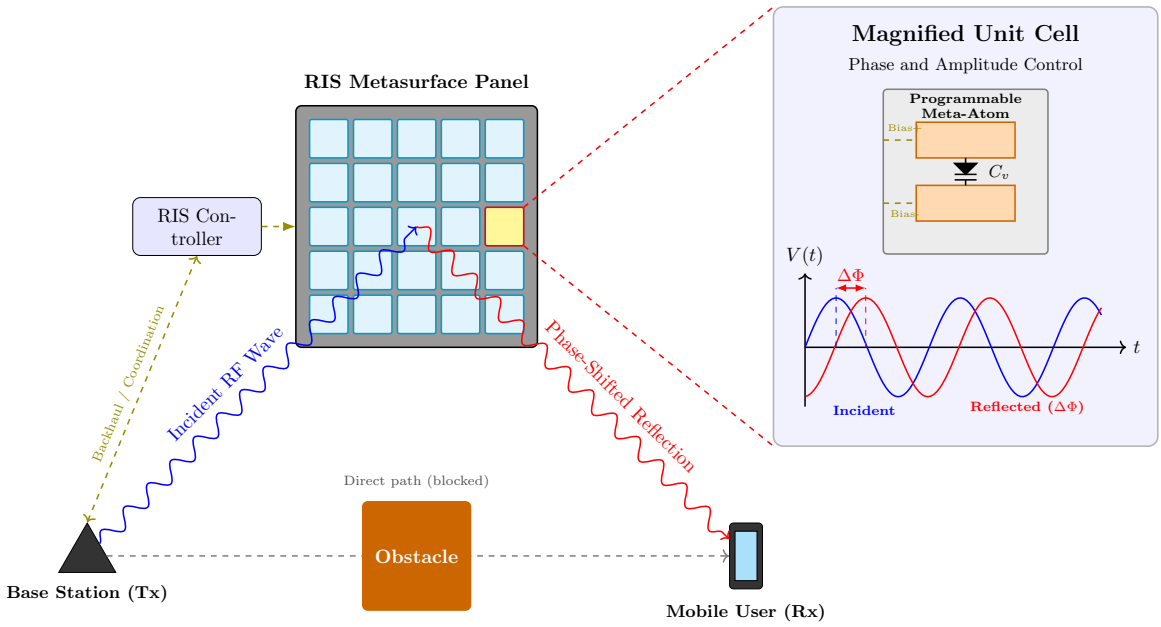


Figure 1 – Schematic representation of an RIS-assisted wireless communication system, illustrating the mitigation of macroscopic shadowing and the establishment of a virtual LoS link between the BS and the UE.

nature of the reflecting elements mandates that their phase shifts be jointly optimized with the active transmissions from the BS and UE to maximize end-to-end communication performance (Wu *et al.*, 2021).

Consequently, this poses a major hurdle, as a fundamental requirement of Beyond 5G (B5G) and 6G networks is the provisioning of URLLC (Hakeem; Hussein; Kim, 2022; Liu *et al.*, 2021). This imposes stringent latency constraints on network control algorithms, necessitating minimal overhead during the RIS reconfiguration phase-specifically regarding the control signaling delays among the UE, the BS, and the RIS controller. Minimizing this reconfiguration overhead is paramount to guarantee high Quality of Service (QoS), particularly for highly mobile users (Liu *et al.*, 2021; Chen *et al.*, 2023).

Compounding this latency challenge is the frequent unavailability of high-rate sensing infrastructure, forcing the RIS control mechanism to rely on stand-alone GNSS measurements for UE localization. However, relying exclusively on GNSS in dense urban environments introduces critical vulnerabilities. First, stand-alone GNSS is highly susceptible to signal interruption, or shading, caused by the same macroscopic urban infrastructure that induces multipath fading (Kaplan; Hegarty, 2017). When the number of visible satellites drops due to blockage from buildings or tunnels, the receiver is forced into a degraded dead-reckoning mode.

Second, a severe temporal mismatch exists between the GNSS update rate

and the wireless channel dynamics. While the propagation channel fluctuates on the order of milliseconds (ElMossallamy *et al.*, 2020), the actual navigation process that outputs the UE coordinates operates in discrete time, typically at just 1 Hz (Kaplan; Hegarty, 2017). Even state-of-the-art commercial high-precision GNSS modules generally reach maximum navigation update rates of only 10 to 20 Hz (ZED-F9P-04B. . . , 2022). Because of this low-frequency, discrete-time nature, if a mobile UE alters its trajectory between updates, simply extrapolating the last known position yields significant estimation errors (Kaplan; Hegarty, 2017).

Because the wireless propagation channel is intrinsically stochastic and highly dynamic, driven by continuous user mobility, transient environmental obstacles, and atmospheric fluctuations (ElMossallamy *et al.*, 2020), relying solely on sparse GNSS updates prevents the RIS from accurately and rapidly tracking the UE. During the prolonged inter-sample periods and shading outages, the system essentially operates with massive spatial blind spots.

This temporal mismatch inevitably leads to severe beam misalignment, drastically degrading beamforming gain and overall communication performance (Chen *et al.*, 2023). Consequently, these compounded hardware and environmental limitations highlight the critical necessity for a robust predictive control strategy capable of estimating UE trajectories in real time.

Recent literature has explored various optimization techniques to address these challenges (He *et al.*, 2020; Zhang; Jiang; Yu, 2023); however, their practical applicability is often bounded by strict power, throughput, or interference constraints (He *et al.*, 2022). Alternatively, data-driven approaches have gained traction. For instance, (Fabiani *et al.*, 2024) proposes a transformer-based architecture fusing camera and GNSS data to predict beam states up to 500 ms in advance, while (Mariotto; Yoma; Almeida, 2025) leverages unsupervised clustering to extract true vehicular trajectories from sparse, noisy Global Positioning System (GPS) data. Despite their efficacy, these machine learning paradigms are intrinsically constrained by their reliance on massive training datasets and bad generalization, hindering their rapid deployment in dynamic, real-world scenarios.

To circumvent the limitations of data-heavy models, the Kalman filter emerges as a highly efficient, model-based alternative. By fusing a kinematic motion model with sequential, noisy observations, the Kalman filter yields optimal recursive state estimates with minimal computational burden, making it exceptionally well-suited for real-time tracking in mobile wireless networks (Särkkä, 2013).

Motivated by these challenges, the present work proposes a novel predictive control framework based on Kalman filtering. By fusing sparse, discrete-time GNSS measurements with a kinematic motion model, the proposed algorithm achieves con-

tinuous, high-precision estimation of the mobile UE's inter-sample spatial states. This approach effectively bridges the temporal update gaps and maintains robust tracking, even during temporary signal outages or periods of high measurement uncertainty. Furthermore, these real-time trajectory predictions facilitate proactive RIS phase reconfiguration, directly compensating for system control latency and preemptively anticipating UE mobility. Consequently, this proactive beam alignment maximizes the received signal power and circumvents the severe performance penalties inherent to traditional, reactive tracking protocols.

2.2 Kalman Filter

The Kalman filter is an optimal recursive algorithm widely utilized for estimating the hidden states of dynamic systems. It operates through a continuous two-stage cycle: prediction and update (Grewal; Andrews, 2008; Särkkä, 2013).

In the prediction step, both the state estimate and its associated uncertainty are propagated from time step $i-1$ to time step i. The system state is assumed to evolve according to a linear dynamic model with additive Gaussian process noise (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\mathbf{s}_i = \mathbf{F}\mathbf{s}_{i-1} + \mathbf{w}_{i-1}, \qquad \mathbf{w}_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{2.1}$$

where $\mathbf{s}_i$ denotes the state vector at time step i, $\mathbf{F}$ is the state-transition matrix, and $\mathbf{w}_{i-1}$ is the process-noise vector with covariance matrix $\mathbf{Q}$, which quantifies the uncertainty introduced by the system dynamics.

Based on this model, the *a priori* state estimate and its corresponding error covariance matrix are obtained as (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\hat{\mathbf{s}}_{i|i-1} = \mathbf{F}\hat{\mathbf{s}}_{i-1|i-1}, \tag{2.2}$$

$$\mathbf{P}_{i|i-1} = \mathbf{F}\mathbf{P}_{i-1|i-1}\mathbf{F}^T + \mathbf{Q}, \tag{2.3}$$

where $\hat{\mathbf{s}}_{i|i-1}$ denotes the predicted state estimate at time step i given the observations up to time step $i-1$, and $\mathbf{P}_{i|i-1}$ is the corresponding *a priori* error covariance matrix, which quantifies the uncertainty associated with the predicted state estimate, including both the uncertainty of each state component and the correlations between estimation errors in different components of the state vector. Similarly, $\mathbf{P}_{i-1|i-1}$ denotes the *a posteriori* error covariance matrix at time step $i-1$, that is, the uncertainty associated with the updated state estimate after incorporating the previous measurement. Finally, $(\cdot)^T$ denotes the transpose of a vector or matrix.

When a new measurement becomes available, the update step incorporates this information to refine the a priori state estimate. This is achieved by comparing the

actual measurement with the predicted measurement obtained from the predicted state estimate. The resulting innovation vector is defined as (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\mathbf{e}_i = \mathbf{r}_i - \mathbf{H}\hat{\mathbf{s}}_{i|i-1}, \tag{2.4}$$

where $\mathbf{r}_i$ denotes the measurement vector, corresponding in this work to the UE position measurement, and $\mathbf{H}$ is the observation matrix that maps the state space to the measurement space. In particular, $\mathbf{H}$ selects and combines the components of the state vector that are directly observable, thereby transforming the predicted state estimate into its corresponding predicted measurement.

The uncertainty associated with the innovation vector is characterized by its covariance matrix, denoted here by $\mathbf{C}_i$, which is given by (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\mathbf{C}_i = \mathbf{H}\mathbf{P}_{i|i-1}\mathbf{H}^T + \mathbf{R}, \tag{2.5}$$

where $\mathbf{R}$ is the measurement-noise covariance matrix, which characterizes the uncertainty affecting the observations as well as possible correlations between measurement errors. Thus, $\mathbf{C}_i$ represents the covariance of the innovation vector and captures the combined effect of the predicted-state uncertainty projected into the measurement space and the uncertainty introduced by the measurement process.

Based on this innovation covariance, the Kalman gain matrix, $\mathbf{K}_i$ is computed as (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\mathbf{K}_i = \mathbf{P}_{i|i-1}\mathbf{H}^T\mathbf{C}_i^{-1}. \tag{2.6}$$

This matrix determines how strongly the predicted state estimate should be corrected using the new measurement. More specifically, it balances the uncertainty in the prediction against the uncertainty in the observation: if the predicted state is highly uncertain relative to the measurement, the correction is larger, whereas if the measurement is more uncertain, the update relies more heavily on the prediction.

The a posteriori state estimate and its corresponding error covariance matrix are then updated as follows (Brown; Hwang, 1996; Grewal; Andrews, 2008; Särkkä, 2013):

$$\hat{\mathbf{s}}_{i|i} = \hat{\mathbf{s}}_{i|i-1} + \mathbf{K}_i\mathbf{e}_i, \tag{2.7}$$

$$\mathbf{P}_{i|i} = (\mathbf{I} - \mathbf{K}_i\mathbf{H})\,\mathbf{P}_{i|i-1}, \tag{2.8}$$

where $\hat{\mathbf{s}}_{i|i}$ denotes the updated state estimate after incorporating the measurement at time step i, $\mathbf{P}_{i|i}$ is the corresponding a posteriori error covariance matrix, and $\mathbf{I}$ is the identity matrix of appropriate dimension. The matrix $\mathbf{P}_{i|i}$ quantifies the remaining uncertainty after the update step and is generally smaller than $\mathbf{P}_{i|i-1}$, reflecting the reduction in uncertainty achieved by incorporating the new measurement.

2.2.1 System Model

In this work, two motion models are considered to describe the UE dynamics: the CA model and the CV model. The corresponding state vectors are defined as:

$$\mathbf{s}_{\mathsf{CA}} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T, \tag{2.9}$$

$$\mathbf{s}_{\mathsf{CV}} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T, \tag{2.10}$$

where x, y, and z denote the UE position coordinates; $\dot{x}$, $\dot{y}$, and $\dot{z}$ denote the corresponding velocity components; and $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ denote the acceleration components, which are included only in the CA model.

The Kalman filter is initialized using the first available position measurement. Since no prior information about the UE motion is assumed to be available at the initial time step, all velocity and acceleration components of the state vector are initialized to zero. This choice provides a neutral initial condition while allowing subsequent measurements to progressively refine the estimated motion state.

The state-transition models adopted in this work are derived from the classical kinematic equations of motion over a sampling interval Δt (Brown; Hwang, 1996; Särkkä, 2013). Under the CA assumption, the state evolution in each spatial coordinate is described by:

$$\begin{aligned} x_i &= x_{i-1} + \dot{x}_{i-1}\Delta t + \frac{1}{2}\ddot{x}_{i-1}\Delta t^2, \quad \dot{x}_i = \dot{x}_{i-1} + \ddot{x}_{i-1}\Delta t, \quad \ddot{x}_i = \ddot{x}_{i-1}, \\ y_i &= y_{i-1} + \dot{y}_{i-1}\Delta t + \frac{1}{2}\ddot{y}_{i-1}\Delta t^2, \quad \dot{y}_i = \dot{y}_{i-1} + \ddot{y}_{i-1}\Delta t, \quad \ddot{y}_i = \ddot{y}_{i-1}, \\ z_i &= z_{i-1} + \dot{z}_{i-1}\Delta t + \frac{1}{2}\ddot{z}_{i-1}\Delta t^2, \quad \dot{z}_i = \dot{z}_{i-1} + \ddot{z}_{i-1}\Delta t, \quad \ddot{z}_i = \ddot{z}_{i-1}. \end{aligned} \tag{2.11}$$

These equations indicate that, under the CA model, the position at time step i depends on both the velocity and acceleration at time step $i-1$, the velocity evolves linearly with acceleration, and the acceleration is assumed to remain constant over the interval Δt. In contrast, under the CV model, the acceleration terms are omitted, so that the position evolves linearly with velocity while the velocity remains constant.

Constant-Acceleration Model

For the 9-dimensional state vector, the state-transition matrix associated with the CA model is given by:

$$\mathbf{F}_{\mathsf{CA}} = \mathbf{I}_3 \otimes \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.12}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix and $\otimes$ denotes the Kronecker product. This block-diagonal structure reflects the assumption that the motion along the x-, y-, and z-axes evolves independently according to the same constant-acceleration kinematic model.

Constant-Velocity Model

For the 6-dimensional state vector, the state-transition matrix associated with the CV model is given by:

$$\mathbf{F}_{\mathsf{CV}} = \mathbf{I}_3 \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}. \tag{2.13}$$

This matrix follows directly from the assumption that the velocity remains constant over the interval Δt, such that the position changes linearly with time in each spatial coordinate.

The matrices $\mathbf{F}_{\mathsf{CA}}$ and $\mathbf{F}_{\mathsf{CV}}$ are employed in the prediction step of the Kalman filter to propagate the state estimate from one time step to the next according to the assumed motion model.

2.2.2 Covariance Matrices

In this work, the initial error covariance matrix and the process-noise covariance matrix are assumed to have a block-diagonal structure. This choice is based on the simplifying assumption that the estimation errors associated with the x-, y-, and z-coordinates are mutually uncorrelated. Accordingly, the uncertainty in each spatial axis is modeled independently, while the coupling among state components within the same axis, such as position, velocity, and acceleration, is preserved. Although cross-axis coupling may arise during turning maneuvers, where the motion along one coordinate can depend on the motion along another, this assumption is exact for nearly rectilinear motion and remains a practical linear approximation in more maneuvering scenarios. The resulting block-diagonal structure simplifies the covariance model, reduces the number of parameters to be specified, and still captures the main uncertainty characteristics of the system, which is why it is commonly adopted in tracking problems.

Error Covariance Matrix

The error covariance matrix characterizes the uncertainty associated with the state estimate and the statistical relationships between the estimation errors of the state components. In the Kalman filter framework, it is defined as (Brown; Hwang, 1996; Grewal; Andrews, 2008):

$$\mathbf{P} = \mathbb{E}\left[(\mathbf{s} - \hat{\mathbf{s}})(\mathbf{s} - \hat{\mathbf{s}})^T\right], \tag{2.14}$$

where s denotes the true state vector, $\hat{\mathrm{s}}$ is its estimate, $\mathbb{E}[\cdot]$ denotes the expectation operator, and $(\cdot)^T$ denotes the transpose of a vector or matrix. Accordingly, each diagonal entry of $\mathbf{P}$ represents the variance of the estimation error associated with a given state component, whereas each off-diagonal entry represents the covariance between the estimation errors of two distinct state components.

For both the CV and CA models, $\mathbf{P}$ is written in block-diagonal form as:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_z \end{bmatrix}, \tag{2.15}$$

where $\mathbf{0}$ denotes a zero matrix of appropriate dimensions, and $\mathbf{P}_x$, $\mathbf{P}_y$, and $\mathbf{P}_z$ describe the uncertainty associated with the state components of the x-, y-, and z-axes, respectively.

Each block $\mathbf{P}_k$, with $k \in \{x, y, z\}$, is a square matrix of dimension $q \times q$, where $q = 2$ for the CV model and $q = 3$ for the CA model. More specifically,

$$\mathbf{P}_k = \begin{cases} \begin{bmatrix} p_{kk} & p_{k\dot{k}} \\ p_{\dot{k}k} & p_{\dot{k}\dot{k}} \end{bmatrix}, & \text{CV model,} \\ \begin{bmatrix} p_{kk} & p_{k\dot{k}} & p_{k\ddot{k}} \\ p_{\dot{k}k} & p_{\dot{k}\dot{k}} & p_{\dot{k}\ddot{k}} \\ p_{\ddot{k}k} & p_{\ddot{k}\dot{k}} & p_{\ddot{k}\ddot{k}} \end{bmatrix}, & \text{CA model.} \end{cases}$$

For the CV model, each block contains the uncertainty associated with position and velocity along a given axis, together with their mutual covariance. For the CA model, each block additionally includes the uncertainty associated with acceleration and its covariance with the position and velocity components. Therefore, the diagonal elements quantify the uncertainty of each individual state component, whereas the off-diagonal elements quantify the correlation between estimation errors of different state components within the same coordinate axis.

At the initial time step, the covariance matrix is typically chosen with relatively large diagonal entries in order to reflect the uncertainty associated with the initial state estimate. This is particularly important for the velocity and acceleration components, which are often initialized without direct measurements and are therefore subject to greater uncertainty than the position components. A larger initial covariance expresses lower confidence in the initial estimate and allows the Kalman filter to adapt more rapidly as new measurements become available.

Process-Noise Covariance Matrix

The process-noise covariance matrix $\mathbf{Q}$ characterizes the uncertainty introduced by the state-transition model. More specifically, it accounts for unmodeled dynamics and random perturbations that are not captured by the deterministic part of the motion model. In the discrete-time Kalman filter, the process noise is modeled as a zero-mean Gaussian random vector:

$$\mathbf{w}_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{2.16}$$

with covariance:

$$\mathbf{Q} = \mathbb{E}\Big[\mathbf{w}_{i-1}\mathbf{w}_{i-1}^{T}\Big]. \tag{2.17}$$

More generally, the discrete-time covariance matrix can be interpreted as the continuous-time process-noise covariance integrated over one sampling interval and propagated through the system dynamics (Grewal; Andrews, 2008).

Consistently with the assumptions adopted for the initial error covariance matrix, $\mathbf{Q}$ is assumed to have block-diagonal form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_z \end{bmatrix}, \tag{2.18}$$

where $\mathbf{Q}_x$, $\mathbf{Q}_y$, and $\mathbf{Q}_z$ denote the process-noise covariance matrices associated with the x-, y-, and z-axes, respectively. This structure reflects the assumption that the random perturbations affecting different spatial coordinates are mutually uncorrelated, while preserving the coupling among position, velocity, and acceleration within each individual axis.

Following the discrete-noise model adopted in Becker (2023), the random input is assumed to be constant over each sampling interval Δt and to change independently from one interval to the next. Under this assumption, the process-noise covariance matrix can be obtained by projecting the variance of the random driving term onto the state vector through the corresponding motion model.

Constant-Velocity Model

For the CV model, the one-dimensional state vector associated with coordinate $k \in \{x, y, z\}$ is:

$$\mathbf{s}_{k,i}^{\mathsf{CV}} = \begin{bmatrix} k_i & \dot{k}_i \end{bmatrix}^{T}.$$

Although the deterministic model assumes constant velocity, the actual motion may still be affected by random acceleration. Let this random acceleration during the interval Δt be denoted by $a_{k,i-1}$, with:

$$a_{k,i-1} \sim \mathcal{N}(0, \sigma_{a,k}^{2}).$$

Its contribution to the state vector over one sampling interval is:

$$\mathbf{w}_{k,i-1}^{\mathsf{CV}} = \mathbf{G}_{\mathsf{CV}}\, a_{k,i-1}, \qquad \mathbf{G}_{\mathsf{CV}} = \begin{bmatrix} \frac{1}{2}\Delta t^{2} \\ \Delta t \end{bmatrix}, \tag{2.19}$$

where $\mathbf{G}_{\mathsf{CV}}$ is the input matrix that maps the acceleration perturbation into the position and velocity states.

The corresponding process-noise covariance block is therefore:

$$\mathbf{Q}_k^{\mathsf{CV}} = \mathbb{E}\Big[\mathbf{w}_{k,i-1}^{\mathsf{CV}} \left(\mathbf{w}_{k,i-1}^{\mathsf{CV}}\right)^T\Big] \tag{2.20}$$

$$= \mathbb{E}\Big[\mathbf{G}_{\mathsf{CV}} a_{k,i-1} a_{k,i-1} \mathbf{G}_{\mathsf{CV}}^T\Big] \tag{2.21}$$

$$= \mathbf{G}_{\mathsf{CV}}\, \mathbb{E}\Big[a_{k,i-1}^2\Big]\, \mathbf{G}_{\mathsf{CV}}^T \tag{2.22}$$

$$= \sigma_{a,k}^2\, \mathbf{G}_{\mathsf{CV}} \mathbf{G}_{\mathsf{CV}}^T. \tag{2.23}$$

Hence,

$$\mathbf{Q}_k^{\mathsf{CV}} = \sigma_{a,k}^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}. \tag{2.24}$$

This is the standard discrete-time process-noise covariance for the constant-velocity model and corresponds to the projection of the acceleration variance through the input matrix (Becker, 2023).

The full covariance matrix for the 6-dimensional CV state vector is then:

$$\mathbf{Q}_{\mathsf{CV}} = \mathbf{I}_3 \otimes \mathbf{Q}_k^{\mathsf{CV}}. \tag{2.25}$$

Constant-Acceleration Model

For the CA model, the one-dimensional state vector associated with coordinate $k \in \{x, y, z\}$ is:

$$\mathbf{s}_{k,i}^{\mathsf{CA}} = \begin{bmatrix} k_i & \dot{k}_i & \ddot{k}_i \end{bmatrix}^T.$$

In this case, acceleration is explicitly included in the state vector. Therefore, the process noise is modeled as a random perturbation acting on the acceleration state. Let the variance of this perturbation be denoted by $\sigma_{a,k}^2$. Following the construction in (Becker, 2023), this uncertainty is represented by:

$$\mathbf{Q}_{a,k} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \sigma_{a,k}^2. \tag{2.26}$$

Using the one-dimensional constant-acceleration state-transition matrix:

$$\mathbf{F}_k^{\mathsf{CA}} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.27}$$

the corresponding process-noise covariance block is obtained as:

$$\mathbf{Q}_k^{\mathsf{CA}} = \mathbf{F}_k^{\mathsf{CA}}\, \mathbf{Q}_{a,k}\, \left(\mathbf{F}_k^{\mathsf{CA}}\right)^T. \tag{2.28}$$

Carrying out the multiplication yields:

$$\mathbf{Q}_k^{\mathsf{CA}} = \sigma_{a,k}^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix}. \tag{2.29}$$

This expression shows that uncertainty injected into the acceleration state propagates to the velocity and position states through the kinematic model, thereby inducing correlations among all three state components (Becker, 2023).

Accordingly, the full process-noise covariance matrix for the 9-dimensional CA state vector is:

$$\mathbf{Q}_{\mathsf{CA}} = \mathbf{I}_3 \otimes \mathbf{Q}_k^{\mathsf{CA}}. \tag{2.30}$$

The diagonal entries of $\mathbf{Q}_k^{\mathsf{CV}}$ and $\mathbf{Q}_k^{\mathsf{CA}}$ quantify the amount of uncertainty injected into each state component, whereas the off-diagonal entries describe the correlations induced by the common random driving term. Consequently, larger values in $\mathbf{Q}$ correspond to lower confidence in the assumed motion model and make the filter more responsive to changes in motion, while smaller values lead to smoother but less adaptive estimates.

2.2.3 Measurement Model

In the proposed system, each measurement corresponds to the UE Cartesian position relative to the RIS. Accordingly, the measurement model relates the observed position vector to the system state through:

$$\mathbf{r}_i = \mathbf{H}\mathbf{s}_i + \mathbf{v}_i, \tag{2.31}$$

where $\mathbf{r}_i$ denotes the measurement vector at time step i, $\mathbf{H}$ is the observation matrix, and $\mathbf{v}_i$ is the measurement-noise vector. Since only the position components are directly observed, the matrix $\mathbf{H}$ maps the full state vector onto the measurement space by selecting the Cartesian coordinates (x, y, z), while the velocity and acceleration components remain unobserved.

For the CA model, the observation matrix is given by:

$$\mathbf{H}_{\mathsf{CA}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{2.32}$$

whereas for the CV model it becomes:

$$\mathbf{H}_{\mathsf{CV}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.33}$$

Assuming that the measurement errors along the three spatial coordinates are mutually uncorrelated, the measurement-noise covariance matrix is modeled as:

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}, \tag{2.34}$$

where σ_x^2, σ_y^2, and σ_z^2 denote the measurement-error variances along the x-, y-, and z-axes, respectively. This diagonal structure reflects the assumption that the uncertainty affecting each coordinate measurement is independent of the others.

2.3 Methodology

The system considered in this paper comprises a fixed BS located at position $[100, -100, 0]$ m, a RIS located at $[0, 0, 0]$ m, and a UE moving within a predefined area. Throughout the UE trajectory, a physical obstacle is assumed to block the LoS path between the BS and the UE, as illustrated in Fig. 2. Both the UE and the BS are assumed to employ isotropic antennas.

To simulate realistic vehicular motion, two trajectories extracted from real-world GPS datasets are considered. Trajectory 1 is taken from (Onyekpe *et al.*, 2021), which provides 10 Hz GPS samples collected from multiple cars and drivers operating on public roads in the United Kingdom, Nigeria, and France. Trajectory 2 is obtained from (Hull, 2024) and consists of 1 Hz tracking data from South African minibus taxis, including dynamic driving behaviors such as aggressive acceleration.

In this work, the recorded GPS positions are treated as the reference ground-truth trajectories for the optimal positioning solution. Since these measurements already reflect realistic vehicular mobility patterns, no additional noise is added to the ground-truth trajectories themselves. For the remaining positioning-related evaluations, measurement uncertainty is modeled by adding zero-mean Gaussian noise with a standard deviation equivalent to $0.5\,\mathrm{m}$. To enable quasi-continuous system evaluation, both trajectories are upsampled to 100 Hz by linear interpolation, which is adequate given the relatively smooth nature of the underlying vehicle motion.

In addition to the empirical trajectories, a synthetic scenario, denoted as Trajectory 3, is evaluated through a Monte Carlo simulation with 100 repetitions. In this case, the UE follows a path generated by a random-walk process in acceleration. At each time step, the position and velocity are updated according to:

$$\begin{cases} \mathbf{u}_t = \mathbf{u}_{t-1} + \dot{\mathbf{u}}_{t-1}\Delta t + \frac{1}{2}\ddot{\mathbf{u}}_t \Delta t^2, \\ \dot{\mathbf{u}}_t = \dot{\mathbf{u}}_{t-1} + \ddot{\mathbf{u}}_t \Delta t, \end{cases} \tag{2.35}$$

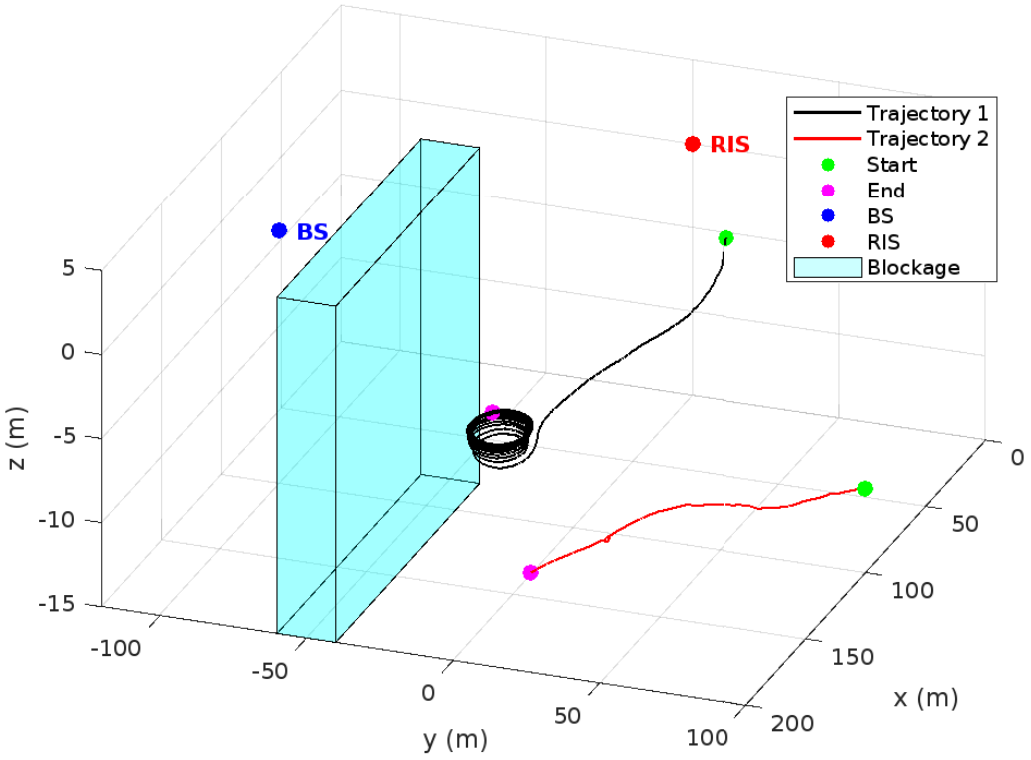


Figure 2 – Trajectories of the UE extracted from real datasets.

where $\mathbf{u}_t$, $\dot{\mathbf{u}}_t$, and $\ddot{\mathbf{u}}_t$ denote the UE position, velocity, and acceleration vectors at time step t, respectively. The acceleration $\ddot{\mathbf{u}}_t$ is independently drawn at each time step from a zero-mean Gaussian distribution with variance $0.1\ \mathrm{m}^2/\mathrm{s}^4$. The initial position and velocity are both set to zero. This formulation allows the generation of a broad range of motion patterns.

The UE trajectory is estimated in real time using two Kalman-filter-based predictors: one based on the CA motion model and the other based on the CV motion model. Both filters operate under a dual-rate structure composed of a high-frequency prediction step and a low-frequency update step. The prediction step is executed every 10 ms, yielding frequent estimates of the UE position, velocity, and acceleration, which are then used to continuously update the RIS configuration for signal reflection. The update step is performed only when a new GPS measurement becomes available, according to the sampling rate of the corresponding dataset. At that point, the filter refines its internal state by incorporating the new measurement and correcting the drift accumulated during prediction.

For each predicted state, the phase shift that maximizes the received power at the UE can be determined for each RIS element as (Costa; Borgese, 2021):

$$\angle\Gamma_{m,n} = \mathrm{mod}\Big(k_0\left(|u^{\mathsf{tx}}_{m,n}| + |\hat{u}^{\mathsf{rx}}_{m,n}|\right), 2\pi\Big), \tag{2.36}$$

where $|u^{\mathsf{tx}}_{m,n}|$ is the exact distance from the BS to the (m,n)-th RIS element, and $|\hat{u}^{\mathsf{rx}}_{m,n}|$ is the estimated distance from the UE to the same element. If $|\hat{u}^{\mathsf{rx}}_{m,n}|$ is accurately estimated, the phase shifts obtained from Eq. (2.36) maximize the received power by promoting fully constructive interference at the UE position.

The channel is modeled using a path-loss-based formulation, in which the received power is given by (Costa; Borgese, 2021):

$$P_r = \frac{P_t \lambda^2}{(4\pi)^2} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{G_t(\theta^{\mathsf{tx}}_{m,n}) G_r(\theta^{\mathsf{rx}}_{m,n}) \sigma(\theta^{\mathsf{tx}}_{m,n}, \theta^{\mathsf{rx}}_{m,n}) |\Gamma_{m,n}|^2}{|r^{\mathsf{tx}}_{m,n}|^2 |r^{\mathsf{rx}}_{m,n}|^2} \times e^{-2j\angle\Gamma_{m,n} + jk_0(|u^{\mathsf{tx}}_{m,n}| + |u^{\mathsf{rx}}_{m,n}|)}. \tag{2.37}$$

Here, P_t denotes the transmit power, λ is the signal wavelength, and G_t and G_r represent the transmit and receive antenna gains, respectively, both taken as unity owing to the assumption of isotropic radiation. The RIS consists of M vertical and N horizontal elements. The angles $\theta^{\mathsf{tx}}_{m,n}$ and $\theta^{\mathsf{rx}}_{m,n}$ denote the angles of incidence and reflection at the (m,n)-th RIS element, respectively, while $|u^{\mathsf{rx}}_{m,n}|$ is the exact distance between the UE and the (m,n)-th RIS element. The radar cross-section of a unit cell, denoted by σ, is given by (Costa; Borgese, 2021):

$$\sigma(\theta^{\mathsf{tx}}_{m,n}, \theta^{\mathsf{rx}}_{m,n}) = 4\pi \left(\frac{D^2}{\lambda}\right)^2 \cos^2(\theta^{\mathsf{tx}}_{m,n}) \tag{2.38}$$

$$\times \left(\frac{\sin\left(\frac{k_0 D}{2}\left(\sin\theta^{\mathsf{rx}}_{m,n} - \sin\theta^{t}_{m,n}\right)\right)}{\frac{k_0 D}{2}\left(\sin\theta^{\mathsf{rx}}_{m,n} - \sin\theta^{\mathsf{tx}}_{m,n}\right)} \right)^2, \tag{2.39}$$

where D is the periodicity of the RIS elements, taken in this work as $\lambda/2$.

Unless otherwise stated, the CA model is adopted as the default state-space formulation for the Kalman filter in the remainder of this paper.

2.4 Numerical Results

This section evaluates the performance of the proposed Kalman-filter-based RIS configuration strategy against a naive baseline, in which the RIS is reconfigured only when a new GPS position measurement becomes available. The comparison is performed in terms of the received power at the UE, considering sampling rates of 1 Hz, 5 Hz, and 10 Hz. The main parameters adopted for the simulations are summarized in Table 1.

Figs. 3, 4, and 5 show the instantaneous received-power degradation at the UE for the three considered trajectories. Each figure compares the naive method, the Kalman filter based on the CV motion model, and the Kalman filter based on the CA motion model under different measurement sampling rates. For Trajectory 3, Fig. 5

Table 1 – Simulation parameters.

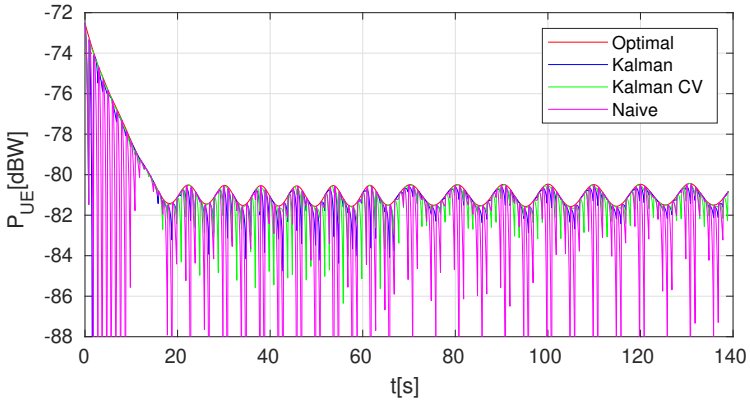
Parameter	**Value**
Carrier frequency	$f = 3.5$ GHz
Wavelength	$\lambda = c/f \approx 85.7$ mm
Transmit power	$P_t = 50$ W
Antenna gains	$G_t = G_r = 1$ (isotropic)
BS position	$[100,\ -100,\ 0]$ m
RIS position	$[0,\ 0,\ 0]$ m
RIS array size	$M \times N = 30 \times 30$
Element periodicity	$D = \lambda/2$
Kalman prediction rate	100 Hz
GPS sampling rates	1, 5, 10 Hz
Trajectory upsampling	100 Hz (linear interpolation)
Monte Carlo runs (Traj. 3)	100
Acceleration variance (Traj. 3)	0.1 m^2/s^4
Positioning Noise	0.5m

corresponds to a single realization of the Monte Carlo simulation. Table 2 reports the average received-power degradation, in dB, for each trajectory, method, and sampling-rate configuration. Since the values represent degradation relative to the ideal case, values closer to 0 dB indicate better performance.

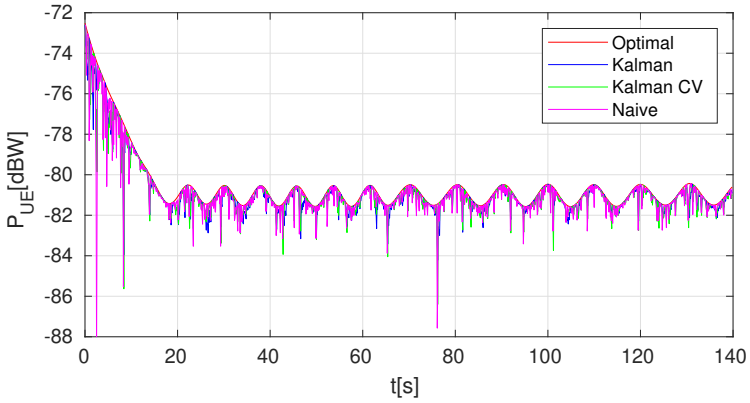
Table 2 – Average received-power degradation [dB] at the UE for different methods, sampling rates, and trajectories (values closer to 0 dB indicate better performance). Best results per trajectory and sampling rate are highlighted in bold.

Trajectory	Method	Sampling Rate (Hz)		
		1Hz	5Hz	10Hz
Trajectory 1	Kalman (CA)	$-$**0.3290**	-0.3462	$-$**0.0454**
	Kalman (CV)	-0.3707	$-$**0.2875**	-0.2542
	Naive	-1.5868	-0.3471	-0.0841
Trajectory 2	Kalman (CA)	-0.3389	$-$**0.0838**	$-$**0.0438**
	Kalman (CV)	$-$**0.3372**	-0.1013	-0.0533
	Naive	-0.6102	-0.1239	-0.1070
Trajectory 3	Kalman (CA)	-0.2341	-0.0516	-0.0357
	Kalman (CV)	$-$**0.2200**	$-$**0.0408**	$-$**0.0270**
	Naive	-0.3429	-0.1615	-0.1644

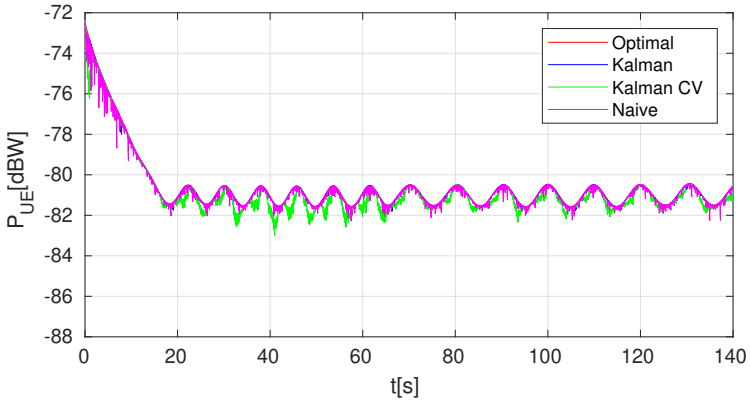
The results show that predictive RIS control based on Kalman filtering consistently outperforms the naive strategy. Across all trajectories and sampling rates, the Kalman-based methods yield average received-power degradation values closer to 0 dB than the baseline. This indicates that estimating the UE motion between measurement updates is effective for preserving RIS phase alignment, especially when the available positioning information is sparse.



(a) Sampling rate of 1 Hz.



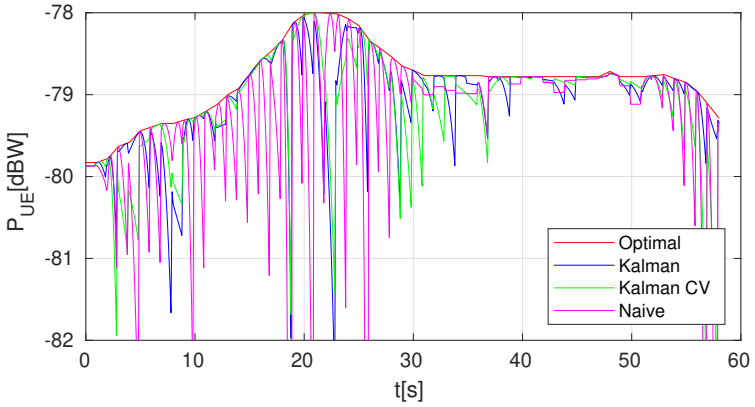
(b) Sampling rate of 5 Hz.



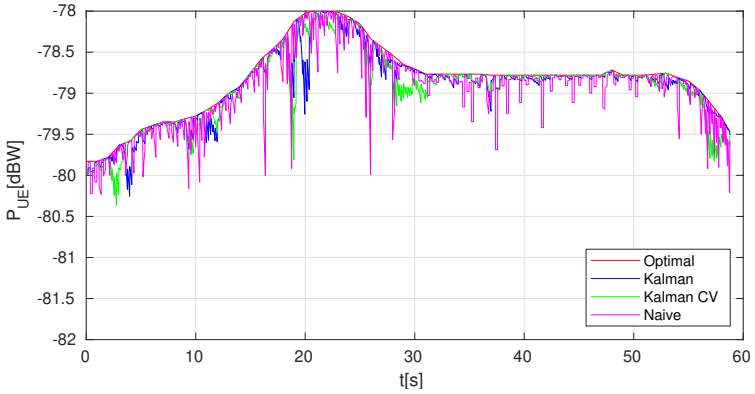
(c) Sampling rate of 10 Hz.

Figure 3 – Instantaneous received-power degradation at the UE for Trajectory 1 under three measurement sampling rates. The results compare the naive RIS reconfiguration strategy with the Kalman-filter-based predictors using the CV and CA motion models.

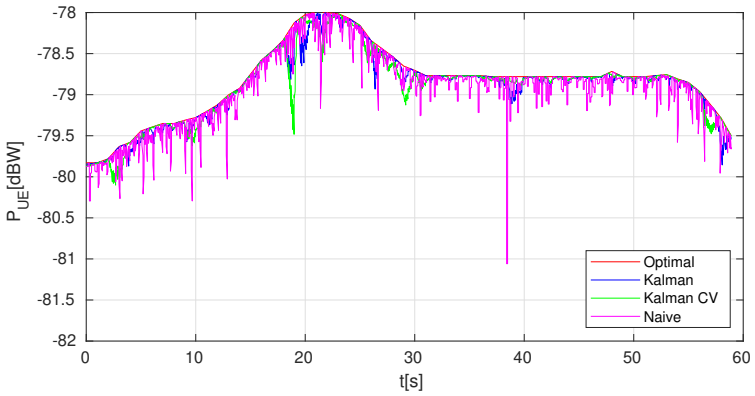
The largest improvements are observed at the lowest sampling rate, namely 1 Hz. In this regime, the naive method relies on increasingly outdated position information between consecutive updates, which leads to a significant mismatch between the actual UE position and the RIS phase configuration. By contrast, the Kalman-based methods propagate the UE state during these intervals and therefore maintain a more accurate estimate of the user position. This effect is particularly evident for Trajectory 1,



(a) Sampling rate of 1 Hz.



(b) Sampling rate of 5 Hz.

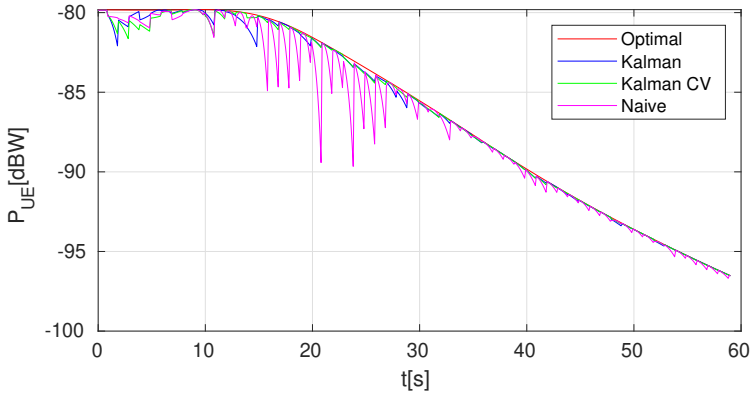


(c) Sampling rate of 10 Hz.

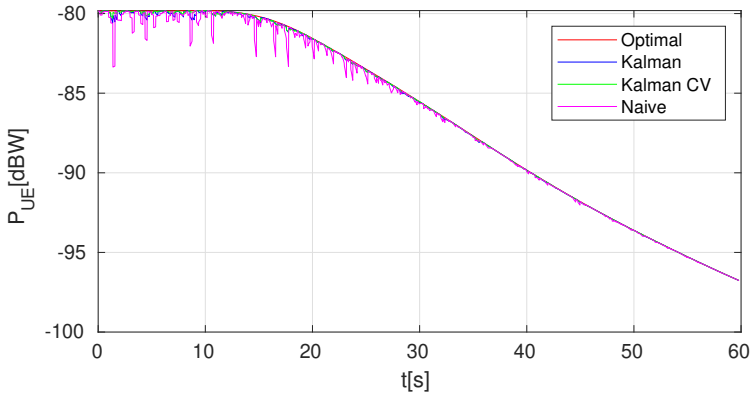
Figure 4 – Instantaneous received-power degradation at the UE for Trajectory 2 under three measurement sampling rates. The predictive methods consistently yield smaller degradation than the naive baseline, particularly at lower sampling rates.

where the CA-based Kalman filter improves the average received-power degradation from -1.5868 dB to -0.3290 dB. This substantial difference highlights the importance of prediction when the measurement rate is low.

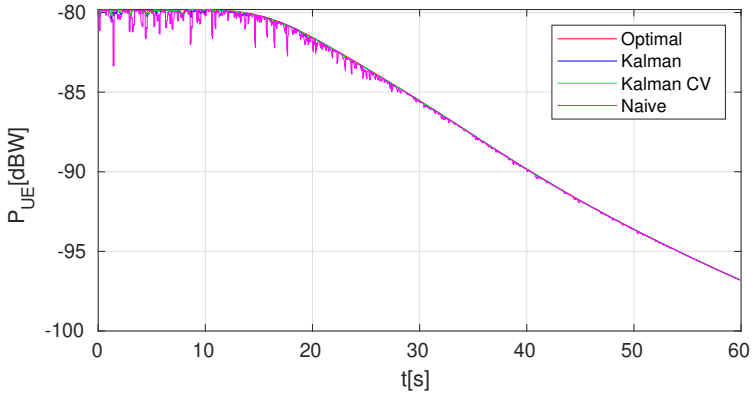
A second clear trend is the influence of the sampling rate itself. For all methods, increasing the measurement frequency from 1 Hz to 5 Hz and 10 Hz generally reduces the received-power degradation. This is expected, since more frequent



(a) Sampling rate of 1 Hz.



(b) Sampling rate of 5 Hz.



(c) Sampling rate of 10 Hz.

Figure 5 – Instantaneous received-power degradation at the UE for Trajectory 3, shown for one realization of the Monte Carlo simulation, under three measurement sampling rates. Both Kalman-filter-based predictors provide more stable and consistently lower degradation than the naive method.

measurements allow the RIS configuration to be updated using more recent position information, thereby limiting the accumulation of tracking error. However, the improvement is more pronounced for the naive method than for the Kalman-based approaches. This behavior arises because the naive strategy depends entirely on the arrival of new measurements, whereas the Kalman filter can partially compensate for sparse updates through model-based prediction. As a result, the relative benefit of predictive tracking is most significant at low sampling rates.

The comparison between the CA and CV Kalman filters further shows that the most suitable motion model depends on the trajectory characteristics. For Trajectory 1, which exhibits stronger curvature and more pronounced directional changes, the CA model achieves the best overall performance, especially at 10 Hz, where the average degradation improves from -0.2542 dB for the CV model to -0.0454 dB for the CA model. This result is consistent with the fact that the CA model explicitly captures acceleration and is therefore better suited to represent nonuniform motion. In contrast, the CV model assumes constant velocity between updates and thus tends to accumulate systematic prediction errors when the trajectory contains turns or abrupt speed variations.

For Trajectories 2 and 3, the difference between the two Kalman models is smaller, and in some cases the CV model slightly outperforms the CA model. This suggests that, for smoother trajectories or for motion with less persistent acceleration, the simpler CV formulation may already provide a sufficiently accurate prediction. In such cases, the additional acceleration states of the CA model do not necessarily translate into a measurable performance gain. This behavior is particularly apparent for the simulated Trajectory 3, in which both Kalman variants remain close to one another while still clearly outperforming the naive baseline.

From a signal-level perspective, the temporal plots reinforce these conclusions. The naive method exhibits larger and more abrupt variations in received-power degradation, especially at low sampling rates, because the RIS configuration remains fixed between measurement updates while the UE continues to move. In contrast, the Kalman-based methods produce smoother temporal profiles and smaller deviations, indicating that the predicted state remains sufficiently accurate to keep the reflected beam better aligned with the UE. The residual fluctuations observed in the Kalman curves can be attributed to the accumulation of prediction errors between updates, as well as to mismatches between the assumed motion model and the actual trajectory.

Overall, these results demonstrate that predictive tracking substantially improves RIS reconfiguration in mobile scenarios. The benefit is most significant when measurements are infrequent and when the motion exhibits time-varying dynamics. Among the two prediction models, the CA formulation is generally more robust in trajectories involving acceleration and turning behavior, whereas the CV model can remain competitive in smoother motion scenarios. Therefore, the proposed Kalman-filter-based framework provides a practical and effective solution for maintaining RIS beam alignment under realistic mobility conditions.

2.5 Conclusions and Future Research Directions

This chapter presented a predictive RIS control strategy based on Kalman filtering to mitigate the limited availability of high-rate UE location information. The proposed approach focuses on scenarios in which the UE position is obtained from low-rate GPS updates, which represents a relevant constraint in practical deployments.

The numerical results showed that Kalman-filter-based prediction can substantially improve RIS reconfiguration performance when compared with a naive update strategy. Across different trajectories and sampling rates, the proposed method reduced the degradation in received power at the UE, with the largest gains observed at low sampling rates. In the most challenging case considered, the predictive approach achieved an improvement of approximately five times relative to the naive baseline. These results highlight the potential of Kalman filtering to preserve beam alignment over longer intervals without fresh position measurements.

The chapter also compared two motion models within the Kalman filtering framework, namely the CA and CV models. The results indicate that the CA model is more suitable for scenarios involving turning behavior or abrupt motion variations, since it explicitly accounts for acceleration. In contrast, the CV model tends to lose accuracy when the UE trajectory deviates from approximately uniform motion, as observed in Trajectory 1.

Despite these promising results, some limitations should be considered. The performance of the Kalman filter depends strongly on the tuning of the process-noise and measurement-noise covariance matrices. Although these matrices are often assumed to be static or perfectly known, this assumption may not hold in realistic B5G/6G scenarios, where mobility patterns, sensing quality, blockage conditions, and propagation dynamics may vary over time. Poor parameter selection may lead to accumulated prediction errors, unstable estimates, or even filter divergence, especially under rapidly changing mobility conditions. Therefore, the robustness of the proposed predictive strategy is closely related to how accurately the estimator represents the uncertainty of the system.

A further limitation is associated with model mismatch. In practice, the UE motion may not follow the assumed CV or CA dynamics, particularly in the presence of abrupt maneuvers, stop-and-go behavior, lane changes, or other context-dependent mobility patterns. The propagation environment may also differ from the simplified path-loss-based formulation adopted in this work. These mismatches can introduce prediction errors that accumulate over time and reduce the effectiveness of the proposed control strategy.

The covariance structure adopted in this chapter also involves simplifications.

The framework assumes that the estimation and process errors associated with the x-, y-, and z-coordinates are mutually uncorrelated, leading to block-diagonal covariance matrices. This assumption is reasonable for approximately rectilinear motion and simplifies the formulation. However, in scenarios with stronger cross-axis coupling, such as turning maneuvers or more complex mobility patterns, neglecting inter-axis correlations may limit the filter's ability to capture the true uncertainty structure of the motion.

It should also be emphasized that the models used in this study are intentionally simplified. The UE motion was represented through CV and CA models, while the propagation environment was described using a basic path-loss formulation with isotropic antennas and an ideal RIS. These assumptions make the problem tractable and allow the core idea to be demonstrated clearly. However, they also imply that the reported gains should be interpreted with caution. In practical deployments, the achievable performance may be affected by irregular UE motion, multipath propagation, hardware impairments, quantized phase shifts, insertion losses, finite switching times, and severe GPS errors, such as outages or large outliers.

The analysis was also limited to a single UE and a restricted set of trajectory profiles. Although the proposed framework provides a useful proof of concept, its extension to dense multi-user scenarios and more diverse mobility conditions remains an open issue. Further investigation is therefore required before the approach can be generalized to realistic B5G/6G deployments.

Future research should further investigate ways to improve the realism, robustness, and scalability of the proposed framework. One promising direction is the development of adaptive Kalman-filter tuning strategies, in which the process-noise and measurement-noise covariance matrices are dynamically adjusted according to mobility patterns, sensing reliability, and prediction residuals. In this context, data-driven methods, such as long short-term memory networks and Transformer-based models, could be explored either to learn suitable covariance parameters from trajectory and measurement data or to model user trajectories directly. By capturing temporal dependencies and complex mobility patterns, these learning-based approaches may improve long-term prediction accuracy, reduce drift accumulation, and mitigate filter divergence under nonstationary operating conditions. Consequently, they may contribute to decreasing the probability of outage events in realistic 6G networks, where user mobility, propagation conditions, and sensing quality may vary significantly over time.

More sophisticated motion models and state-estimation techniques could also be investigated. For example, the Interacting Multiple Model Kalman Filter (IMM-KF) can combine multiple motion models and is better suited to systems with switching dynamics. Machine-learning-based predictors may also be useful in scenarios with highly nonlinear or context-dependent motion patterns. In parallel, more realistic channel and

RIS models should be incorporated to better capture practical propagation effects and hardware limitations.

An additional research direction is the translation of the positioning and beam-alignment results into traditional communication-level metrics. For instance, the temporal degradation of the RIS pointing accuracy between two consecutive sensor updates could be mapped into received-power loss, signal-to-noise ratio degradation, achievable-rate reduction, or outage probability. This would enable a more direct comparison between the proposed predictive strategy and the naive baseline from the perspective of communication reliability. In particular, it would be useful to evaluate how the accumulated beam misalignment before the next sensor update affects the probability of outage, especially at low positioning update rates.

Finally, future work should consider approaches that are robust to positioning anomalies. Sensor-fusion strategies combining GPS with additional sources of information, such as inertial sensors, could improve reliability under degraded positioning conditions. Filtering methods capable of handling model mismatch and correlated uncertainties more explicitly may also reduce the impact of the simplifying assumptions adopted in this chapter. Moving beyond numerical simulations toward hardware-in-the-loop experiments and real-world prototyping will be essential to validate the practical benefits of predictive RIS control and to identify implementation challenges that cannot be fully captured in simulation.

In summary, this work provides an initial step toward predictive RIS control for mobile scenarios. The results confirm the potential of Kalman-filter-based prediction to improve beam alignment under limited positioning update rates. At the same time, they show that further advances are needed in adaptive filter calibration, communication-level performance evaluation, robustness, modeling realism, uncertainty representation, and multi-user scalability before the full practical potential of this approach can be realized.

3 DEEP UNFOLDED SCA FOR ISAC NON-CONVEX PRECODING DESIGN: LEARNING ADAPTIVE SCHEDULES

3.1 Background and Motivation

ISAC has emerged as a key enabling technology for next-generation wireless networks, offering more efficient spectrum utilization by integrating radar sensing and data transmission within a shared hardware and spectral platform (Liu *et al.*, 2020, 2022b; Liu *et al.*, 2022a). Early studies established the potential of such dual-function designs, showing that joint use of spectral and hardware resources can improve overall system efficiency while alleviating spectrum congestion (Paul; Chiriyath; Bliss, 2017; Liu *et al.*, 2018; Cheng *et al.*, 2019). However, achieving high performance in ISAC systems remains challenging because radar sensing and wireless communication generally impose competing design objectives (Paul; Chiriyath; Bliss, 2017; Liu *et al.*, 2018, 2020; Liu *et al.*, 2022a). In multi-antenna deployments, these objectives are largely governed by how transmit energy is distributed across space, since the same waveform must illuminate sensing targets while also delivering reliable signals to communication users. For this reason, beamforming has become a fundamental design tool in ISAC, enabling the transmitted signal to be spatially shaped so as to form desired radar beampatterns while simultaneously satisfying stringent communication QoS requirements (Liu *et al.*, 2017; Hua; Xu; Han, 2021, 2023).

Beyond the beamforming mechanism itself, a central question is how sensing performance should be quantified within the beamformer design. While early approaches often relied on beampattern matching, recent literature has increasingly adopted optimization frameworks based on minimization of the CRB, which provides a more rigorous measure of sensing accuracy (Liu *et al.*, 2022a; Liu *et al.*, 2022c,d; Cheng *et al.*, 2019). Beamforming designs optimized with respect to the CRB have been shown to achieve superior target estimation performance compared with conventional beampattern-matching strategies (Liu *et al.*, 2022c,d; Cheng *et al.*, 2019).

However, the resulting optimization problems are inherently non-convex and mathematically intricate because the sensing objective is tightly coupled with the per-user SINR constraints. To address this challenge, prior work developed a rigorous framework based on SCA (Liu *et al.*, 2022c,d), in which the non-convex terms are iteratively approximated through first-order Taylor expansions. This methodology has proven highly adaptable, effectively handling coupled constraints in more advanced scenarios such as full-duplex ISAC systems (He *et al.*, 2023) and wideband hybrid beamforming architectures (Nguyen *et al.*, 2023). By decomposing the original prob-

lem into a sequence of tractable Second-Order Cone Program (SOCP)s, the resulting algorithm can be solved efficiently and converges to a stationary point under standard assumptions (Liu *et al.*, 2022d).

Empirical studies have consistently demonstrated the effectiveness of SCA-based methods across a range of ISAC settings. In foundational narrowband designs, SCA has been shown to achieve performance close to Semidefinite Relaxation (SDR) benchmarks while substantially improving the CRB relative to beampattern-based baselines, with relatively fast convergence (Liu *et al.*, 2022d). In wideband hybrid architectures, combining SCA with manifold optimization has yielded notable sum-rate gains under stringent sensing constraints while reducing the number of required Radio Frequency (RF) chains (Nguyen *et al.*, 2023). Related convex approximation frameworks have also proven effective in full-duplex ISAC, where they improve spectral and power efficiency by mitigating self-interference and handling coupled uplink–downlink constraints (He *et al.*, 2023).

Despite these strong results, optimization-based beamforming methods remain computationally demanding, which limits their suitability for real-time deployment in fast-varying ISAC environments. This challenge becomes even more pronounced in scenarios involving wideband channels, large-scale antenna arrays, or hybrid analog–digital architectures, where repeated iterative optimization can incur substantial latency and computational overhead (Nguyen *et al.*, 2024; Temiz *et al.*, 2025; Zhang *et al.*, 2025a; Gao *et al.*, 2026).

A natural alternative is to leverage machine learning to approximate the beamforming solution through data-driven inference. However, conventional deep learning approaches often provide limited interpretability and weak theoretical guarantees (Deka *et al.*, 2025; He *et al.*, 2019). Moreover, as largely black-box models, architectures such as Convolutional Neural Network (CNN)s and Multilayer Perceptron (MLP)s typically do not exploit the underlying signal model or optimization structure of the beamforming problem (Hershey; Le Roux; Weninger, 2014; He *et al.*, 2019; Deka *et al.*, 2025; Zhang *et al.*, 2025a). As a result, they generally require large volumes of labeled data to learn the channel-to-precoder mapping and may generalize poorly when operating conditions deviate from the training distribution (Gregor; LeCun, 2010; He *et al.*, 2019; Zhang *et al.*, 2025a).

To bridge the gap between principled optimization and low-latency inference, DU has emerged as a promising framework for ISAC beamforming (Balatsoukas-Stimming; Studer, 2019; Zhang *et al.*, 2025a; Zhang *et al.*, 2025c). Rather than learning an unconstrained mapping from inputs to beamformers, DU constructs a trainable neural architecture by unfolding the iterations of a model-based optimization algorithm. In doing so, it preserves the structure, domain knowledge, and interpretability of the

underlying physical model while introducing a limited number of trainable parameters as a strong inductive bias (He *et al.*, 2019; Hershey; Le Roux; Weninger, 2014; Balatsoukas-Stimming; Studer, 2019; Monga; Li; Eldar, 2021). Because the learning process is restricted to a small set of algorithm-related parameters, such as step sizes or penalty coefficients, DU can offer improved data efficiency, interpretability, and inference speed relative to both purely optimization-based solvers and fully black-box learning methods (Zhang *et al.*, 2025a; Temiz *et al.*, 2025).

3.1.1 Related Works

Following its success in broader wireless optimization tasks, such as Intelligent Reflecting Surface (IRS) beamforming (Liu *et al.*, 2022e) and power allocation (Li; Verma; Segarra, 2023), DU has increasingly been applied to the non-convex design problems arising in ISAC. Existing works span several important directions. In hybrid beamforming, the deep unfolded Projected Gradient Descent (PGA) framework in Nguyen *et al.* (2024) maps iterative projection updates into trainable network layers, enabling efficient handling of the coupled analog–digital constraints and achieving strong spectral-efficiency performance with substantially fewer iterations than conventional manifold optimization. For IRS-aided ISAC systems, Zhang *et al.* (2025b) proposed an unfolded Alternating Direction Method of Multipliers (ADMM)-based architecture for joint active and passive beamforming, in which costly matrix inversion steps are replaced by linear approximations with trainable parameters, thereby improving computational efficiency.

Recent studies have also extended deep unfolding to robust ISAC transceiver design under Channel State Information (CSI) uncertainty. For example, (Zhang *et al.*, 2025c) showed that unfolding iterative optimization procedures can significantly reduce inference latency while improving robustness to channel imperfections relative to conventional data-driven baselines (Zhang *et al.*, 2025a; Temiz *et al.*, 2025). These developments indicate that deep unfolding is not limited to a single architecture or constraint set, but can be adapted to a broad range of structured ISAC optimization problems.

Particularly relevant to the present work is the extension of deep unfolding to algorithms based on SCA. The foundational study in Liu *et al.* (2022e) demonstrated that unfolding stochastic SCA iterations for IRS-assisted full-duplex systems can replace computationally expensive matrix operations with learnable first-order approximations, thereby preserving the core iterative structure of the original solver while substantially reducing complexity. This result provides an important methodological basis for developing low-latency neural architectures that retain the principled structure of SCA-based optimization.

3.1.2 Contributions

Motivated by recent advances in ISAC optimization, this work builds on the CRB-minimization framework developed in Liu *et al.* (2022c,d). We consider the joint design of communication and sensing precoders to minimize radar-channel estimation error while strictly satisfying communication QoS requirements. Although SCA provides a principled way to handle the resulting non-convex problem through a sequence of convex approximations, its computational burden remains a major obstacle for latency-sensitive ISAC deployments (Zhang *et al.*, 2025a). In particular, each iteration requires solving subproblems whose cost scales poorly with the problem dimension; for the CRB-driven objective in Eq. (3.14), gradient evaluation involves inversion of the precoding covariance matrix, resulting in a complexity of $\mathcal{O}(N_t^3)$ with respect to the number of transmit antennas (Golub; Van Loan, 2013). This difficulty is further exacerbated by sequential backtracking procedures, such as Armijo line search, which hinder parallel execution, as well as by the sensitivity of first-order methods to ill-conditioned optimization landscapes (Bertsekas, 1999; Nocedal; Wright, 2006; Ruder, 2016; Boyd; Vandenberghe, 2004).

To address these limitations, we propose a model-driven DU framework that unfolds the gradient-based proximal SCA procedure into a trainable architecture for CRB-driven ISAC beamforming. Specifically, we develop a two-level unfolding design that maps both the outer proximal updates and the inner gradient-based steps into a fixed-depth computational graph, enabling end-to-end learning of optimization parameters tailored to the underlying ISAC problem structure. Instead of relying on fixed, hand-crafted hyperparameters, the proposed framework learns adaptive quantities such as step sizes, momentum coefficients, and penalty weights directly from data, thereby improving convergence speed, stability, and feasibility. To further remove a major computational bottleneck of conventional SCA, we replace backtracking line search with a differentiable residual-update mechanism based on sigmoid gating, which is more amenable to parallel execution on modern hardware. Through extensive simulations, we show that the resulting Deep Unfolded SCA (DU-SCA) framework substantially improves feasibility and computational efficiency relative to conventional iterative baselines. To the best of our knowledge, this is the first work to investigate the unfolding of a gradient-based SCA scheme for CRB minimization under joint SINR and transmit-power constraints in ISAC.

3.2 System Model

We consider a downlink ISAC system in which a dual-functional BS is equipped with N_t transmit antennas and $N_r \geq N_t$ receive antennas. The BS serves K single-antenna communication users in a multi-user Multiple-Input Single-Output (MISO) down-

link while simultaneously performing monostatic radar sensing, as illustrated in Fig. 6.

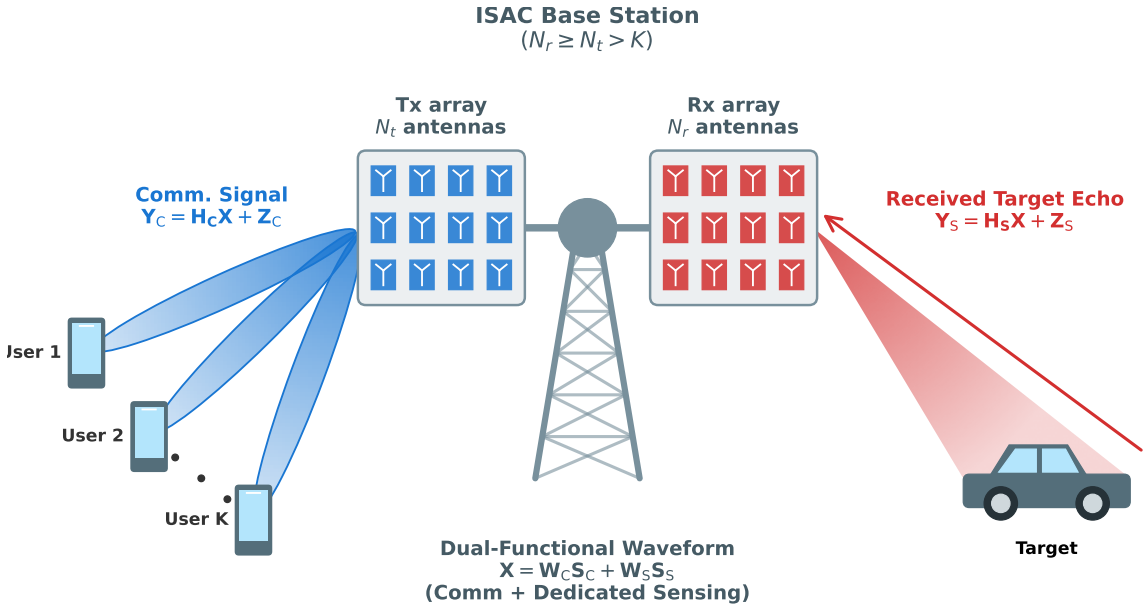


Figure 6 – Illustration of the joint radar-communication downlink system.

3.2.1 Communication Signal Model

The dual-functional waveform transmitted by the BS is denoted by $\mathbf{X} \in \mathbb{C}^{N_t \times L_{\text{frame}}}$, where L_{frame} denotes the frame length. The transmit signal is subject to a total power constraint $\|\mathbf{X}\|_F^2 \leq P_{\text{T}}$, where P_{T} is the maximum transmit power budget.

The received signal at the K communication users is modeled as (Liu *et al.*, 2022c,d):

$$\mathbf{Y}_{\text{C}} = \mathbf{H}_{\text{C}}\mathbf{X} + \mathbf{Z}_{\text{C}}, \tag{3.1}$$

where $\mathbf{H}_{\text{C}} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]^H \in \mathbb{C}^{K \times N_t}$ denotes the aggregate downlink channel matrix, assumed to be perfectly known at the BS. The noise matrix $\mathbf{Z}_{\text{C}} \in \mathbb{C}^{K \times L_{\text{frame}}}$ is modeled as Additive White Gaussian Noise (AWGN), whose entries are independently distributed as $\mathcal{CN}(0, \sigma_{\text{C}}^2)$.

3.2.2 Sensing Signal Model

Concurrently, the echo signal reflected by the target and received at the BS sensing array is modeled as (Liu *et al.*, 2022c):

$$\mathbf{Y}_{\text{S}} = \mathbf{H}_{\text{S}}\mathbf{X} + \mathbf{Z}_{\text{S}}, \tag{3.2}$$

where $\mathbf{H}_{\text{S}} \in \mathbb{C}^{N_r \times N_t}$ denotes the sensing channel, or equivalently the target response matrix, to be estimated. The additive noise term $\mathbf{Z}_{\text{S}} \in \mathbb{C}^{N_r \times L_{\text{frame}}}$ is modeled as having

independently distributed entries drawn from $\mathcal{CN}(0, \sigma_{\mathsf{S}}^2)$.[1]

3.2.3 Radar Estimation Performance

We assume that the BS estimates $\mathbf{H}_{\mathsf{S}}$ from the noisy observation $\mathbf{Y}_{\mathsf{S}}$ using the Maximum Likelihood Estimator (MLE), given by (Liu *et al.*, 2022c,d,b):

$$\hat{\mathbf{H}}_{\mathsf{S}} = \mathbf{Y}_{\mathsf{S}}\mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}. \tag{3.3}$$

Under the assumed additive Gaussian noise model, the estimation performance is characterized by the CRB, which in this setting coincides with the Mean Squared Error (MSE) achieved by the MLE (Liu *et al.*, 2022c,d,b):

$$\mathsf{CRB}(\mathbf{X}) = \mathbb{E}\left\{\|\mathbf{H}_{\mathsf{S}} - \hat{\mathbf{H}}_{\mathsf{S}}\|_F^2\right\} = \frac{\sigma_{\mathsf{S}}^2 N_r}{L_{\mathsf{frame}}} \operatorname{tr}\left(\mathbf{R}_{\mathsf{X}}^{-1}\right), \tag{3.4}$$

where $\mathbf{R}_{\mathsf{X}} = \frac{1}{L_{\mathsf{frame}}}\mathbf{X}\mathbf{X}^H$ denotes the sample covariance matrix of the transmitted waveform.

This expression reveals a fundamental sensing-communication trade-off. In particular, minimizing the sensing error requires $\mathbf{R}_{\mathsf{X}}$ to be invertible, which in turn requires $\mathbf{X}$ to be full row rank, i.e., $\operatorname{rank}(\mathbf{X}) = N_t$. Otherwise, the Fisher information matrix becomes singular, and unbiased estimation of the target response is no longer possible (Liu *et al.*, 2022b).

By contrast, in conventional multi-user MISO downlink transmission, the BS typically sends only K independent data streams. Hence, when $N_t > K$, as is common in massive Multiple-Input Multiple-Output (MIMO) regimes, the transmitted signal satisfies $\operatorname{rank}(\mathbf{X}) \leq K < N_t$, which makes $\mathbf{R}_{\mathsf{X}}$ rank-deficient. As a result, standard communication-centric signaling is generally incompatible with reliable target estimation based on the MLE in Eqs. (3.2)–(3.4) (Hua; Xu; Han, 2021; Liu *et al.*, 2022b).

3.2.4 Precoding Design with Dedicated Sensing Signals

To overcome the rank-deficiency issue discussed in the previous subsection, we adopt the common strategy of introducing $N_t - K$ auxiliary sensing streams, as in Hua, Xu, and Han (2021) and Liu *et al.* (2022c,d,b). These additional streams do not carry communication information and are introduced to enrich the spatial degrees of freedom of the transmitted waveform, thereby enabling a full-rank transmit covariance without compromising the communication signal structure.

[1] Following a common assumption in monostatic/full-duplex ISAC systems, self-interference leakage from the transmit array is assumed to be perfectly suppressed through isolation and cancellation techniques (Liu *et al.*, 2022b; He *et al.*, 2023).

The transmitted waveform is modeled as the superposition of communication-precoded data streams and dedicated sensing signals (Liu *et al.*, 2022d,b):

$$\mathbf{X} = \mathbf{W}_{\mathsf{C}}\mathbf{S}_{\mathsf{C}} + \mathbf{W}_{\mathsf{S}}\mathbf{S}_{\mathsf{S}}, \tag{3.5}$$

where $\mathbf{W}_{\mathsf{C}} \in \mathbb{C}^{N_t \times K}$ and $\mathbf{W}_{\mathsf{S}} \in \mathbb{C}^{N_t \times (N_t - K)}$ denote the communication and sensing precoding matrices, respectively. The symbol matrices $\mathbf{S}_{\mathsf{C}} \in \mathbb{C}^{K \times L_{\text{frame}}}$ and $\mathbf{S}_{\mathsf{S}} \in \mathbb{C}^{(N_t - K) \times L_{\text{frame}}}$ are assumed to be uncorrelated and normalized such that

$$\mathbb{E}[\mathbf{S}_{\mathsf{C}}\mathbf{S}_{\mathsf{C}}^H] = \mathbf{I}_K, \qquad \mathbb{E}[\mathbf{S}_{\mathsf{S}}\mathbf{S}_{\mathsf{S}}^H] = \mathbf{I}_{N_t - K}, \qquad \mathbb{E}[\mathbf{S}_{\mathsf{C}}\mathbf{S}_{\mathsf{S}}^H] = \mathbf{0}.$$

This signal construction enables the transmitted waveform to achieve full spatial rank, which is essential for obtaining an invertible transmit covariance and, consequently, for improving sensing estimation performance.

3.2.5 Impact on Communication SINR

Although the dedicated sensing component $\mathbf{W}_{\mathsf{S}}\mathbf{S}_{\mathsf{S}}$ improves radar estimation capability, it also introduces additional interference to the communication users. Accordingly, the resulting SINR for user k is given by (Hua; Xu; Han, 2021; Liu *et al.*, 2022b,c; Hua; Xu; Han, 2023):

$$\gamma_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\underbrace{\sum_{i \neq k} |\mathbf{h}_k^H \mathbf{w}_i|^2}_{\text{MUI}} + \underbrace{\mathbf{h}_k^H \mathbf{W}_{\mathsf{S}} \mathbf{W}_{\mathsf{S}}^H \mathbf{h}_k}_{\text{Sensing interference}} + \sigma_{\mathsf{C}}^2}, \quad \forall k \in \mathcal{K}, \tag{3.6}$$

where $\mathcal{K} = \{1, \ldots, K\}$ denotes the set of communication users, and $\mathbf{w}_k$ is the k-th column of $\mathbf{W}_{\mathsf{C}}$. The denominator accounts for multi-user interference, sensing-induced interference, and receiver noise. The resulting design problem is therefore to jointly optimize $\mathbf{W}_{\mathsf{C}}$ and $\mathbf{W}_{\mathsf{S}}$ so as to minimize the sensing error, characterized through the CRB, while satisfying the per-user SINR requirements.

3.3 Optimization Problem Formulation

The goal of the considered ISAC system is to jointly design the communication and sensing precoders, collected in the dual-functional matrix:

$$\mathbf{W} = [\mathbf{W}_{\mathsf{C}}, \mathbf{W}_{\mathsf{S}}] \in \mathbb{C}^{N_t \times N_t},$$

so as to improve sensing accuracy while guaranteeing the required communication QoS.

3.3.1 Problem Statement

As established in Liu *et al.* (2022c), under the Gaussian linear observation model, the MSE of the unbiased MLE coincides with the CRB. Therefore, minimizing the sensing estimation error is equivalent to minimizing the CRB, which depends on the inverse transmit covariance matrix.

Moreover, when the frame length L_{frame} is sufficiently large, the sample covariance matrix of the transmitted waveform can be approximated by its statistical counterpart. Under the assumed normalization and mutual uncorrelatedness of the communication and sensing symbol streams, this yields:

$$\mathbf{R}_{\mathsf{X}} = \frac{1}{L_{\text{frame}}}\mathbf{X}\mathbf{X}^H \approx \mathbf{W}\mathbf{W}^H. \tag{3.7}$$

Accordingly, we define the CRB-motivated objective:

$$\mathcal{L}_{\text{CRB}}(\mathbf{W}) \triangleq \mathrm{tr}\Big((\mathbf{W}\mathbf{W}^H)^{-1}\Big), \tag{3.8}$$

which serves as the optimization surrogate of the CRB expression in Eq. (3.4).

The joint precoder design problem is then formulated as (Liu *et al.*, 2022c,d) :

$$\begin{aligned}
\min_{\mathbf{W}} \quad & \mathcal{L}_{\text{CRB}}(\mathbf{W}) && \text{(3.9a)}\\
\text{s.t.} \quad & \gamma_k \geq \Gamma_k, \quad \forall k \in \mathcal{K}, && \text{(3.9b)}\\
& \|\mathbf{W}\|_F^2 \leq P_T. && \text{(3.9c)}
\end{aligned}$$

Here, P_T denotes the maximum transmit-power budget, and Γ_k is the required SINR threshold for user k. Since the objective in Eq. (3.9a) involves the inverse of $\mathbf{W}\mathbf{W}^H$, feasible solutions must yield a full-rank transmit covariance matrix. The problem in Eq. (3.9) therefore captures the fundamental ISAC trade-off: improving radar estimation accuracy through CRB minimization while maintaining the communication quality required by all users.

3.3.2 Differentiable Cost Function Formulation

The optimization problem in Eqs. (3.9a)–(3.9c) includes hard, non-convex constraints, which make direct gradient-based learning difficult (Hua; Xu; Han, 2021; Liu *et al.*, 2022d; Nguyen *et al.*, 2024). To enable DU, we replace these constraints with differentiable penalty terms based on the squared ReLU function, which acts as a one-sided quadratic penalty (Boyd; Vandenberghe, 2004; Nocedal; Wright, 2006). This choice has two useful properties. First, it provides unilateral penalization through the projection operator $[z]^+ \triangleq \max(0, z)$, so that satisfied constraints incur zero penalty and the optimization focuses only on violated constraints (Nocedal; Wright, 2006). Second, the

squared penalty $P(z) = ([z]^+)^2$ is continuously differentiable (C^1) over $\mathbb{R}$, which leads to stable gradients near the constraint boundary while imposing increasingly strong penalties on larger violations (Nocedal; Wright, 2006; Bertsekas, 1999; Boyd; Vandenberghe, 2004).

Accordingly, we define the following unconstrained composite loss:

$$\mathcal{L}_{\text{total}}(\mathbf{W}) = \mathcal{L}_{\text{CRB}}(\mathbf{W}) + \lambda_{\text{p}}\mathcal{L}_{\text{pwr}}(\mathbf{W}) + \lambda_{\text{s}}\mathcal{L}_{\text{sinr}}(\mathbf{W}), \tag{3.10}$$

where λ_{p} and λ_{s} are non-negative hyperparameters that balance sensing performance against constraint satisfaction. Here, $\mathcal{L}_{\text{CRB}}$ promotes accurate sensing by minimizing the radar estimation error bound, while $\mathcal{L}_{\text{pwr}}$ and $\mathcal{L}_{\text{sinr}}$ penalize violations of the transmit-power and communication-QoS constraints, respectively.

The power-constraint penalty is defined as:

$$\mathcal{L}_{\text{pwr}}(\mathbf{W}) = \left[\mathrm{ReLU}\Big(\|\mathbf{W}\|_F^2 - P_T\Big)\right]^2, \tag{3.11}$$

which is activated only when the total transmit power exceeds the budget P_T.

To handle the minimum-SINR constraints, we adopt the Second-Order Cone (SOC)-based reformulation in Liu *et al.* (2022c,d). By exploiting the phase ambiguity of the precoders, the desired signal term can be taken to be real and non-negative, which yields the following differentiable penalty:

$$\mathcal{L}_{\text{sinr}}(\mathbf{W}) = \sum_{k=1}^{K}\left[\mathrm{ReLU}\left(\sqrt{\Gamma_k}\,\|\tilde{\mathbf{h}}_k\|_2 - \sqrt{1+\Gamma_k}\,\Re\{\mathbf{h}_k^H\mathbf{w}_k\}\right)\right]^2, \tag{3.12}$$

where $\|\tilde{\mathbf{h}}_k\|_2 = \|[\mathbf{h}_k^H\mathbf{W},\, \sigma_C]\|_2$ denotes the Euclidean norm of the augmented interference-plus-noise vector.

3.4 Successive Convex Approximation (SCA)

The optimization problem in Eq. (3.9) seeks to minimize the CRB-driven objective:

$$\mathcal{L}_{\text{CRB}}(\mathbf{W}) = \mathrm{tr}\Big((\mathbf{W}\mathbf{W}^H)^{-1}\Big),$$

which is non-convex with respect to the precoding matrix $\mathbf{W}$ (Liu *et al.*, 2022c,d). To address this difficulty, we adopt a proximal SCA strategy (Scutari *et al.*, 2014; Liu *et al.*, 2022d). The key idea is to iteratively replace the original non-convex problem with a sequence of strongly convex surrogate subproblems, each constructed around the current iterate. Under standard assumptions, this procedure yields convergence to a stationary point of the original problem (Sun; Babu; Palomar, 2017; Liu; Lau; Kananian, 2019; Liu *et al.*, 2022c).

3.4.1 Linearization and Gradient Calculation

At the t-th outer iteration, we linearize $\mathcal{L}_{\mathsf{CRB}}(\mathbf{W})$ around the current point $\mathbf{W}^{(t-1)}$. For notational convenience, we define:

$$\mathbf{G}^{(t-1)} \triangleq \nabla \mathcal{L}_{\mathsf{CRB}}(\mathbf{W}^{(t-1)}).$$

Using complex matrix calculus, the gradient is given by (Liu *et al.*, 2022d,c):

$$\mathbf{G}^{(t-1)} = -\left(\mathbf{W}^{(t-1)}(\mathbf{W}^{(t-1)})^H\right)^{-1}\left(\mathbf{W}^{(t-1)}(\mathbf{W}^{(t-1)})^H\right)^{-H}\mathbf{W}^{(t-1)}. \tag{3.13}$$

By defining the covariance matrix at iteration $t-1$ as:

$$\mathbf{R}_{\mathsf{X}}^{(t-1)} \triangleq \mathbf{W}^{(t-1)}(\mathbf{W}^{(t-1)})^H,$$

the gradient can be equivalently expressed as:

$$\mathbf{G}^{(t-1)} = -\left((\mathbf{R}_{\mathsf{X}}^{(t-1)})^{-1}\right)^2\mathbf{W}^{(t-1)}.$$

3.4.2 Convex Subproblem Formulation

To ensure strong convexity of each surrogate problem and stabilize the update trajectory, we introduce the proximal regularization term:

$$\frac{\rho}{2}\|\mathbf{W}-\mathbf{W}^{(t-1)}\|_F^2,$$

where $\rho > 0$ is the proximal parameter (Scutari *et al.*, 2014; Parikh; Boyd, 2014). This term acts as a soft trust-region mechanism by discouraging excessively large deviations from the point where the linearization is constructed (Parikh; Boyd, 2014).

The resulting convex subproblem at iteration t is defined through the surrogate objective:

$$\begin{aligned}\hat{\mathcal{L}}^{(t)}(\mathbf{W}) = {} & \Re\left\{\mathrm{tr}\left((\mathbf{G}^{(t-1)})^H(\mathbf{W}-\mathbf{W}^{(t-1)})\right)\right\} \\ & + \frac{\rho}{2}\|\mathbf{W}-\mathbf{W}^{(t-1)}\|_F^2 \\ & + \lambda_{\mathsf{p}}\mathcal{L}_{\mathsf{pwr}}(\mathbf{W}) + \lambda_{\mathsf{s}}\mathcal{L}_{\mathsf{sinr}}(\mathbf{W}).\end{aligned} \tag{3.14}$$

The first term is the first-order approximation of the non-convex CRB objective, the second term provides proximal regularization, and the last two terms preserve the differentiable penalties associated with the power and SINR constraints.

3.4.3 Optimization via Nesterov Accelerated Gradient Descent

To solve the strongly convex subproblem in Eq. (3.14) efficiently, we employ Nesterov Aceelerated Gradient Descent (NAGD) in its velocity-based form (Sutskever

et al., 2013). This method evaluates the gradient at a look-ahead point and uses momentum to accelerate convergence, which is particularly beneficial in ill-conditioned optimization landscapes.

Let $\mathbf{V}^{(l)}$ denote the velocity matrix, initialized to $\mathbf{0}$, and let $\mathbf{W}^{(l)}$ denote the inner-loop optimization variable at step l. Given momentum coefficient μ and step size η, the look-ahead point is computed as:

$$\tilde{\mathbf{W}}^{(l)} = \mathbf{W}^{(l)} + \mu \mathbf{V}^{(l)}. \tag{3.15}$$

Next, for each training sample i, we evaluate the gradient of the surrogate loss at the extrapolated point:

$$\mathbf{g}_i^{(l)} \triangleq \nabla_{\mathbf{W}} \hat{\mathcal{L}}_i^{(t)}(\tilde{\mathbf{W}}^{(l)}).$$

To improve numerical stability during training, we apply per-sample gradient clipping (Pascanu; Mikolov; Bengio, 2013):

$$\bar{\mathbf{g}}_i^{(l)} = \min\left(1, \frac{C}{\|\mathbf{g}_i^{(l)}\|_F + \epsilon}\right) \mathbf{g}_i^{(l)}, \tag{3.16}$$

where C is the clipping threshold and $\epsilon > 0$ is a small constant introduced for numerical stability. The clipped gradients are then averaged over a mini-batch of size B, yielding:

$$\bar{\mathbf{G}}^{(l)} = \frac{1}{B} \sum_{i=1}^{B} \bar{\mathbf{g}}_i^{(l)}. \tag{3.17}$$

The velocity and optimization variable are then updated as:

$$\mathbf{V}^{(l+1)} = \mu \mathbf{V}^{(l)} - \eta \bar{\mathbf{G}}^{(l)}, \tag{3.18a}$$

$$\mathbf{W}^{(l+1)} = \mathbf{W}^{(l)} + \mathbf{V}^{(l+1)}. \tag{3.18b}$$

This inner optimization is repeated for a fixed number of iterations L, or until the gradient norm falls below a prescribed tolerance.

3.4.4 Step Size Selection via Armijo Rule

Let $\mathbf{W}^\star$ denote the optimal solution of the convex subproblem in Eq. (3.14). Within the SCA framework, this solution defines the descent direction:

$$\mathbf{D}^{(t)} = \mathbf{W}^\star - \mathbf{W}^{(t-1)}$$

for the outer update (Liu *et al.*, 2022d). Although $\mathbf{D}^{(t)}$ is guaranteed to decrease the surrogate objective, taking an excessively large or excessively small step along this direction may fail to provide sufficient decrease in the original non-convex objective, which can lead to instability or slow progress (Bertsekas, 1999; Nocedal; Wright, 2006).

To address this issue, the step size $\alpha \in (0, 1]$ is commonly selected using the Armijo backtracking rule (Bertsekas, 1999; Nocedal; Wright, 2006). This procedure enforces a sufficient-decrease condition relative to the directional derivative and is widely used to promote monotonic descent and convergence to a stationary point in non-convex optimization (Armijo, 1966; Nocedal; Wright, 2006).

Specifically, the algorithm searches for the largest $\alpha \in \{\beta^0, \beta^1, \beta^2, \dots\}$, where $\beta \in (0, 1)$ is the backtracking factor, such that the Armijo condition:

$$\begin{aligned} \mathcal{L}_{\text{total}}(\mathbf{W}^{(t-1)} + \alpha \mathbf{D}^{(t)}) \leq \mathcal{L}_{\text{total}}(\mathbf{W}^{(t-1)}) \\ + c\,\alpha\,\Re\Big\{\Big\langle \nabla \mathcal{L}_{\text{total}}(\mathbf{W}^{(t-1)}), \mathbf{D}^{(t)} \Big\rangle\Big\}, \end{aligned} \tag{3.19}$$

is satisfied, where $c \in (0, 1)$ is the sufficient-decrease parameter. Once α is determined, the outer iterate is updated as (Liu *et al.*, 2022c,d):

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \alpha \mathbf{D}^{(t)}. \tag{3.20}$$

3.5 Proposed Deep Unfolding-based Precoding

To alleviate the computational bottlenecks of traditional SCA solvers, we propose a model-driven DU-SCA framework. As illustrated in Fig. 7, the iterative procedure of the proximal SCA algorithm is unfolded into a fixed-depth computational graph with T cascaded layers, where each layer corresponds to one outer iteration of the original algorithm. In this way, the iterative optimization process is recast as a feed-forward architecture that combines the structure and interpretability of model-based optimization with the low-latency inference of deep learning.

The proposed DU-SCA architecture explicitly embeds the analytical structure of the optimization problem in Eq. (3.10) as an inductive bias. In particular, key algorithmic hyperparameters, including step sizes, momentum coefficients, and penalty weights, are treated as trainable variables rather than fixed design constants. Through offline training, the network learns layer-wise update rules adapted to the statistical distribution of the channel realizations (Monga; Li; Eldar, 2021; Gregor; LeCun, 2010). As a result, the unfolded network can approximate the behavior of the original iterative solver using a substantially smaller number of layers than the number of iterations typically required by conventional SCA (Monga; Li; Eldar, 2021; Gregor; LeCun, 2010). By shifting the main adaptation effort to the offline training stage, the proposed framework enables fast online inference, making it well suited to latency-sensitive ISAC applications (Monga; Li; Eldar, 2021).

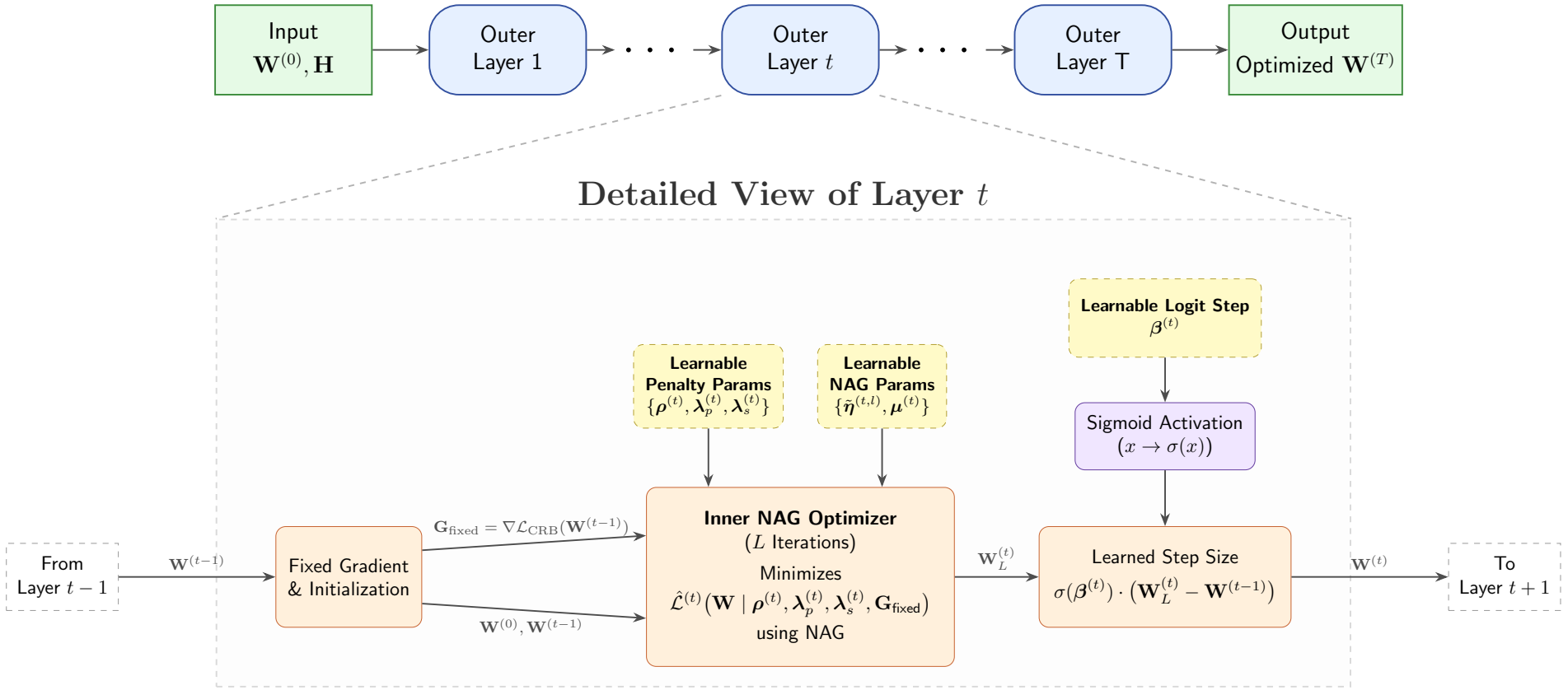


Figure 7 – Topology of the proposed deep unfolded SCA network. The architecture consists of T outer layers, where each layer t performs a convex approximation step followed by an inner optimization loop with learnable parameters.

3.5.1 Adaptive Weighting Mechanism

A major limitation of classical SCA is its dependence on fixed hyperparameters, whose values are typically selected through extensive manual tuning (Balatsoukas-Stimming; Studer, 2019). In practice, this calibration is often costly and problem dependent, since finite-iteration performance is highly sensitive to the choice of proximal and penalty parameters, while general closed-form selection rules are rarely available (Balatsoukas-Stimming; Studer, 2019). As a result, conventional solvers often rely on heuristic choices or conservative bounds that may generalize poorly across different channel conditions.

This sensitivity is particularly pronounced in penalty-based formulations (Deka *et al.*, 2025; Balatsoukas-Stimming; Studer, 2019). If the penalty weights are too small, constraint violations may persist and the resulting precoders may be infeasible. Conversely, overly large weights can lead to ill-conditioning and unstable optimization trajectories, which slow convergence and degrade numerical robustness (Bertsekas, 1999; Nocedal; Wright, 2006; Deka *et al.*, 2025). In ISAC, these effects are especially important because poor parameter choices can distort the balance between sensing accuracy and communication performance (Nguyen *et al.*, 2024; Deka *et al.*, 2025).

To address this issue, the proposed DU-SCA architecture adopts an adaptive weighting mechanism in which the main hyperparameters are learned and allowed to vary across layers. To enforce physical validity, namely non-negativity of the proximal and penalty weights, we parameterize them in the log-domain (Sutton, 1992). Let:

$$\boldsymbol{\Theta}^{(t)} = \left\{ \tilde{\rho}^{(t)}, \tilde{\lambda}_{\mathsf{p}}^{(t)}, \tilde{\lambda}_{\mathsf{s}}^{(t)} \right\}$$

denote the trainable hyperparameters associated with layer t. The corresponding physical parameters used by the unfolded solver are then obtained through the exponential mapping:

$$\rho^{(t)} = e^{\tilde{\rho}^{(t)}}, \quad \lambda_{\mathsf{p}}^{(t)} = e^{\tilde{\lambda}_{\mathsf{p}}^{(t)}}, \quad \lambda_{\mathsf{s}}^{(t)} = e^{\tilde{\lambda}_{\mathsf{s}}^{(t)}}. \tag{3.21}$$

3.5.1.1 Role of the Proximal Weight

The proximal weight $\rho^{(t)}$ controls the effective trust region of the update at layer t (Parikh; Boyd, 2014; Scutari *et al.*, 2014):

- **Conservative updates (large $\rho^{(t)}$):** A larger proximal weight keeps the solution closer to the linearization point $\mathbf{W}^{(t-1)}$, which improves stability when the local approximation is less accurate (Parikh; Boyd, 2014; Scutari *et al.*, 2014);
- **Aggressive updates (small $\rho^{(t)}$):** A smaller proximal weight allows larger deviations from $\mathbf{W}^{(t-1)}$, which can accelerate convergence when the convex surrogate provides a reliable local approximation (Parikh; Boyd, 2014; Scutari *et al.*, 2014).

3.5.1.2 Role of the Penalty Weights

The penalty weights $\lambda_{\mathsf{p}}^{(t)}$ and $\lambda_{\mathsf{s}}^{(t)}$ regulate the trade-off between minimizing the sensing objective and enforcing the power and SINR constraints (Nocedal; Wright, 2006; Shi; Hong, 2020; Shi *et al.*, 2020; Balatsoukas-Stimming; Studer, 2019):

- **Relaxed regime (small penalty weights):** Smaller penalty values place greater emphasis on objective minimization while allowing temporary constraint violations, which may help the optimization trajectory explore the solution space more freely (Shi; Hong, 2020; Shi *et al.*, 2020);
- **Enforcement regime (large penalty weights):** Larger penalty values impose a stronger cost on infeasible solutions, steering the iterates toward the feasible set and improving satisfaction of the physical power and SINR constraints (Shi; Hong, 2020; Shi *et al.*, 2020).

By learning these weights across layers, the network can adapt its optimization trajectory over time, taking more exploratory steps in the early stages and more conservative, feasibility-oriented steps in later layers, which can improve both convergence behavior and final solution quality (Balatsoukas-Stimming; Studer, 2019).

3.5.2 Inner Loop Implementation

Each outer layer t aims to solve the convex subproblem in Eq. (3.14). To do so efficiently, we unfold the solver into L iterations of NAGD (Sutskever *et al.*, 2013). Unlike classical implementations, which typically rely on fixed or heuristically decaying update schedules, the proposed architecture equips the inner solver with learnable control parameters (Balatsoukas-Stimming; Studer, 2019).

Specifically, for each inner iteration $l \in \{1, \ldots, L\}$ within outer layer t, we assign a dedicated step size $\eta^{(t,l)}$ and momentum coefficient $\mu^{(t,l)}$. These parameters are learned independently across layers and inner iterations, allowing the unfolded network to discover a flexible, non-uniform optimization schedule. In this way, the solver can adapt its update dynamics not only across the outer SCA procedure, but also to the local curvature of each intermediate surrogate subproblem.

To ensure positive step sizes, we parameterize $\eta^{(t,l)}$ in the log-domain. Let $\tilde{\eta}^{(t,l)}$ denote the corresponding trainable parameter. The effective step size is then given by:

$$\eta^{(t,l)} = e^{\tilde{\eta}^{(t,l)}}. \tag{3.22}$$

This mapping guarantees $\eta^{(t,l)} > 0$ throughout training, while allowing the learned step sizes to vary over several orders of magnitude.

Similarly, to constrain the momentum coefficient to the stable range $(0, 1)$, we adopt a sigmoid parameterization. Let $\tilde{\mu}^{(t,l)}$ be the trainable variable associated with the l-th inner iteration of layer t. The corresponding momentum coefficient is defined as:

$$\mu^{(t,l)} = \sigma(\tilde{\mu}^{(t,l)}) = \frac{1}{1 + e^{-\tilde{\mu}^{(t,l)}}}. \tag{3.23}$$

3.5.3 Outer Loop Update

Upon completion of the L inner iterations, the unfolded inner solver produces a refined candidate solution, denoted by $\mathbf{W}^{\star}$. In the classical SCA framework (Liu *et al.*, 2022c,d), this candidate is incorporated into the outer iterate through a backtracking line-search procedure, such as the Armijo rule, in order to guarantee sufficient decrease. However, this process is computationally expensive because it requires repeated evaluations of the objective and constraints across multiple backtracking steps, thereby introducing a serial bottleneck that is poorly suited to low-latency deployment (Zhang *et al.*, 2025a; Bertsekas, 1999).

To remove this overhead, the proposed DU-SCA framework replaces the iterative line search with a residual update governed by a learned mixing coefficient. This mechanism allows the network to determine, in a single forward step, how much of the refined candidate should be incorporated into the next outer iterate, thereby balancing stability and progress without repeated objective evaluations (Balatsoukas-Stimming; Studer, 2019). Specifically, we introduce a learnable scalar $\beta^{(t)}$, and define the outer update as:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \sigma(\beta^{(t)})\Big(\mathbf{W}^{\star} - \mathbf{W}^{(t-1)}\Big), \tag{3.24}$$

where $\sigma(\cdot)$ denotes the sigmoid function, which constrains the mixing coefficient to the interval $(0, 1)$.

3.5.4 DU-SCA Training

The proposed DU-SCA architecture is trained end-to-end using an unsupervised learning strategy. Unlike supervised approaches that require pre-computed optimal labels, the network is trained directly by minimizing the physics-based objective derived from the underlying optimization problem. Specifically, the training loss is given by $\mathcal{L}_{\text{total}}$ in Eq. (3.10), which combines the sensing objective with penalty terms associated with power and SINR constraint violations. During training, λ_{pwr} and λ_{sinr} are treated as fixed hyperparameters that assign a large cost to infeasible solutions.

The network parameters are optimized using the Adaptive Moment Estimation (Adam) algorithm, which adapts the learning rate of each parameter based on estimates of the first and second moments of the gradients (Kingma; Ba, 2015). To improve numerical stability across the unfolded layers, we apply gradient clipping to limit

the norm of the backpropagated gradients and thereby mitigate exploding-gradient effects (Pascanu; Mikolov; Bengio, 2013). In addition, a learning-rate annealing schedule is employed, whereby the learning rate is reduced by a fixed factor every N epochs to support more stable and fine-grained convergence during the later stages of training (Bengio, 2012).

The complete forward propagation procedure of the proposed architecture is summarized in Algorithm 1.

Algorithm 1 Deep Unfolded SCA Forward Pass

Require: Channel $\mathbf{H}$, Init $\mathbf{W}^{(0)}$, Layers T, Inner Steps L
Ensure: Optimized Precoder $\mathbf{W}^{(T)}$
for $t = 1$ to T **do**
  $\mathbf{G}_{\text{fixed}} \leftarrow \nabla \mathcal{L}_{\text{CRB}}(\mathbf{W}^{(t-1)})$
  Inner Init: $\mathbf{W}_0 \leftarrow \mathbf{W}^{(t-1)}$, $\mathbf{V} \leftarrow \mathbf{0}$
  / Map log-parameters to physical weights
  $\rho \leftarrow e^{\tilde{\rho}^{(t)}}, \quad \lambda_{\text{p}} \leftarrow e^{\tilde{\lambda}_{\text{p}}^{(t)}}, \quad \lambda_{\text{s}} \leftarrow e^{\tilde{\lambda}_{\text{s}}^{(t)}}$
  for $l = 1$ to L **do**
    / Nesterov Momentum
    $\mu \leftarrow \sigma(\mu^{(t)})$
    $\mathbf{W}_{\text{lookahead}} \leftarrow \mathbf{W}_{l-1} + \mu \cdot \mathbf{V}$
    / Compute Inner Gradients
    $\mathbf{g} \leftarrow \nabla_{\mathbf{W}} \hat{\mathcal{L}}^{(t)}(\mathbf{W}_{\text{lookahead}} \mid \rho, \lambda_{\text{p}}, \lambda_{\text{s}})$
    Clip $\|\mathbf{g}\|$ to G_{clip}
    / Update Velocity and Position
    $\eta \leftarrow e^{\tilde{\eta}_l^{(t)}}$
    $\mathbf{V} \leftarrow \mu \cdot \mathbf{V} - \eta \cdot \mathbf{g}$
    $\mathbf{W}_l \leftarrow \mathbf{W}_{l-1} + \mathbf{V}$
  end for
  $\mathbf{W}_{\text{star}} \leftarrow \mathbf{W}_L$
  / Outer Update with Learned Step
  $\beta \leftarrow \sigma(\beta^{(t)})$
  $\mathbf{W}^{(t)} \leftarrow \mathbf{W}^{(t-1)} + \beta \cdot (\mathbf{W}_{\text{star}} - \mathbf{W}^{(t-1)})$
end for
return $\mathbf{W}^{(T)}$

3.6 Numerical Results

In this section, we evaluate the performance of the proposed deep unfolded joint radar-communication precoding framework. Unless otherwise stated, all reported results are averaged over $N_{\text{test}} = 5000$ independent Monte Carlo channel realizations to ensure statistical reliability.

3.6.1 Simulation Setup

Following the setup in Liu *et al.* (2022c,d), we consider a downlink MIMO-ISAC system with $N_t = 16$ transmit antennas serving K single-antenna users, together with a co-located sensing receiver equipped with $N_r = 20$ antennas. The frame length is fixed at $L_{\text{frame}} = 30$. An AWGN model is assumed, and the noise power at both the communication users and the sensing receiver is set to $\sigma_{\text{C}}^2 = \sigma_{\text{S}}^2 = 0$ dBm (1 mW). The total transmit power budget is fixed at $P_T = 30$ dBm (1 W), corresponding to a transmit Signal-to-Noise Ratio (SNR) of 30 dB. The communication channels are modeled as independent Rayleigh fading vectors, i.e., $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_t})$.

To assess the effectiveness of the proposed method, we compare it against the classical iterative SCA solver described in Section 3.4. Table 3 summarizes the system and algorithmic parameters. Unless otherwise stated, the default setting adopts the bold values in the table, namely $K = 10$ and $\Gamma = 15$ dB, consistent with Liu *et al.* (2022d). This corresponds to a moderately loaded regime ($K/N_t \approx 0.6$) under stringent QoS requirements, and therefore provides a challenging benchmark for joint sensing-communication precoder design.

The simulation framework is calibrated to ensure physical relevance, numerical stability, and fair computational comparison. In particular, the moderately loaded regime ($N_t = 16$, $K = 10$) is chosen to stress interference management and avoid overly simple operating conditions. The penalty weights are selected empirically to balance the sensing objective and feasibility terms, thereby compensating for the different scales of the objective and constraints. In addition, to isolate algorithmic efficiency under latency-sensitive conditions, both the baseline solver and the proposed DU-SCA architecture are restricted to the same total budget of 100 inner optimization steps, so that performance differences reflect learned acceleration rather than unequal computational effort.

3.6.1.1 Justification of Parameter Selection

The simulation parameters in Table 3 were selected according to three main considerations: physical relevance, numerical stability, and computational fairness.

- **System loading ($N_t = 16, K = 10$).** We adopt the configuration in Liu *et al.* (2022d) to enable a fair comparison with established ISAC beamforming benchmarks. In addition, serving $K = 10$ users with $N_t = 16$ transmit antennas yields a moderately loaded system, with load factor $K/N_t \approx 0.6$. This regime is particularly relevant because it places meaningful stress on interference management and precoder design. By contrast, lightly loaded scenarios (e.g., $K = 2$) are

Table 3 – Simulation Parameters and Algorithm Configuration

Parameter	Value / Range
System Model and Default Scenario	
Transmit antennas (N_t)	16
Receive antennas (N_r)	20
Number of users (K)	$\{4, 8, \mathbf{10}, 12, 14\}$
Target SINR (Γ)	$\{5, 10, \mathbf{15}, 20, 25\}$ dB
Transmit power budget (P_T)	$\mathbf{30}$ dBm (1.0 W)
Noise power ($\sigma_{\mathsf{C}}^2, \sigma_{\mathsf{S}}^2$)	$\mathbf{0}$ dBm (1.0 mW)
Channel model	Rayleigh fading, $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_t})$
Frame length (L_{frame})	30 time slots
CSI assumption	Perfect CSI
Classical SCA Baseline	
Outer $\times$ inner iterations	10×10 (total 100)
Step size (η)	1×10^{-3}
Armijo backtracking factor (β)	1×10^{-3}
Nesterov momentum (μ)	0.9
Penalty weights ($\lambda_{\mathsf{p}}, \lambda_{\mathsf{s}}$)	5×10^4 (fixed)
Proposed DU-SCA	
Unrolling depth	$T = 10$ outer layers
Inner steps per layer	$L = 10$
Training samples (N_{train})	1,000 (unsupervised)
Training loss weights	Same as baseline
Learnable parameters	$\eta^{(t,l)}, \mu^{(t,l)}, \rho^{(t)}, \lambda_{\mathsf{p}}^{(t)}, \lambda_{\mathsf{s}}^{(t)}, \beta^{(t)}$

Note: Bold values indicate the default simulation settings.

considerably easier and can often be handled effectively by conventional linear precoding methods.

- **Penalty weights ($\lambda_{\mathsf{p}}, \lambda_{\mathsf{s}}$).** The penalty weights are set to 5×10^3, based on manual tuning through grid-search simulations. This calibration is necessary to compensate for the scale mismatch between the CRB-driven objective and the feasibility terms associated with the power and SINR constraints. If the penalty weights are too small, the solver may reduce the sensing objective at the cost of violating the physical constraints; if they are too large, the optimization may become overly stiff and numerically difficult (Bertsekas, 1999).

- **Computational budget and latency (100 total iterations).** To ensure a fair efficiency comparison, both the classical solver and the proposed DU-SCA architecture are restricted to the same total computational budget of 100 algorithmic steps. Specifically, the baseline uses 10 outer iterations and 10 inner iterations, while the unfolded network uses $T = 10$ layers with $L = 10$ inner steps per layer. This choice reflects two considerations: first, practical ISAC deployments often require precoding under tight latency constraints, especially in fast-varying environments; second, matching the total iteration budget ensures that any performance gain can be attributed to the learned update mechanism rather than to a larger computational allowance.

It is worth noting that, due to computational-time constraints, the hyperparameters of the classical SCA algorithm were tuned using a rapid grid search over a limited set of configurations. Therefore, its performance could potentially be further improved through a more exhaustive tuning procedure tailored to each specific scenario.

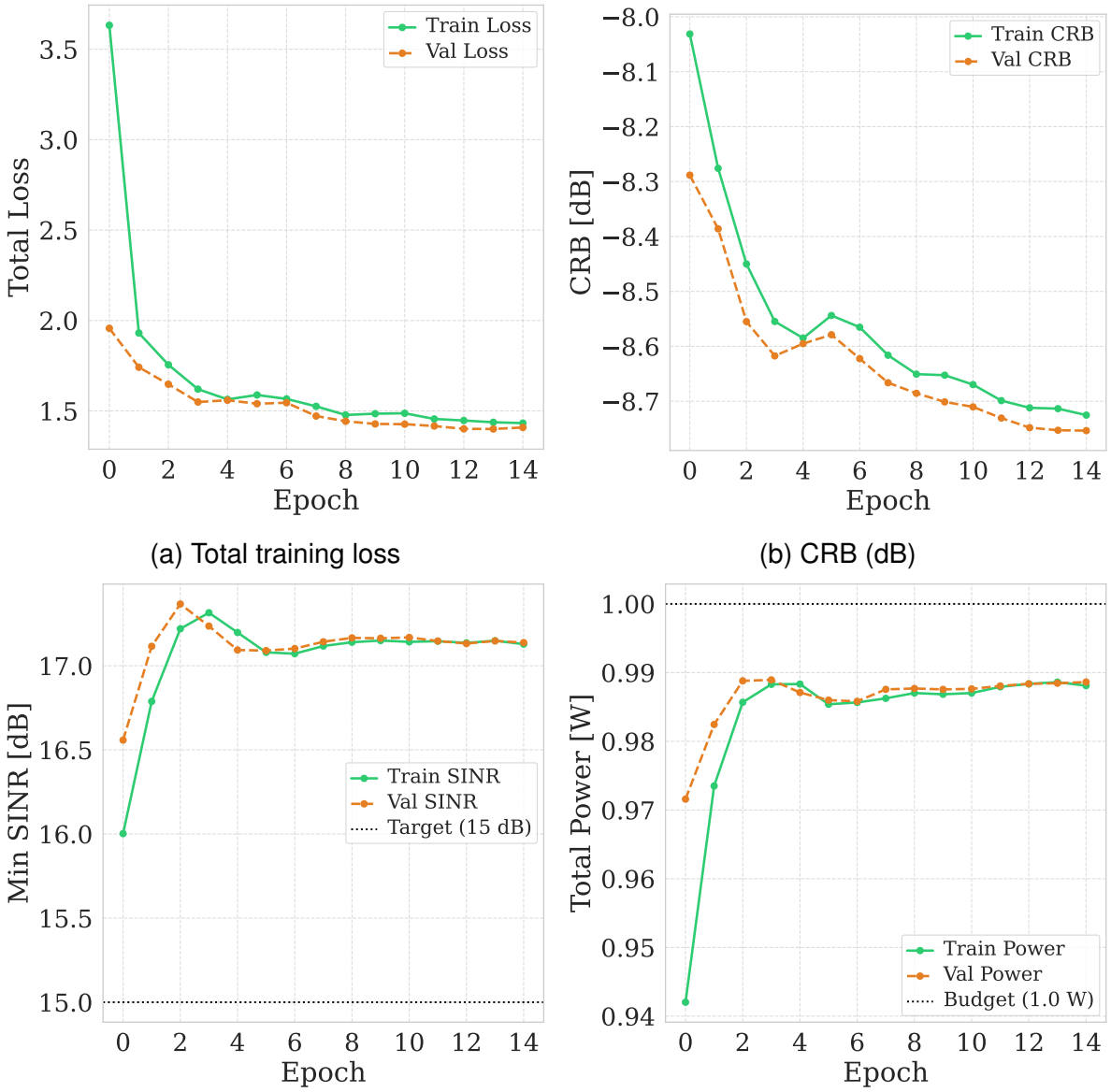
3.6.2 Training Convergence and Adaptive Mechanisms

The proposed DU-SCA architecture replaces manual hyperparameter tuning with a learned adaptive update mechanism. Fig. 8 illustrates the evolution of the training and validation metrics across epochs, where solid and dashed curves denote the training and validation sets, respectively. Two main observations can be drawn from these results.

First, the validation curves closely track the training curves throughout the learning process, indicating good generalization to unseen channel realizations and no evident overfitting, despite the relatively small training set size ($N_{\text{train}} = 1000$). Second, the network rapidly learns to satisfy the feasibility requirements while continuing to improve the sensing objective. In particular, the SINR and power-violation terms are driven to zero within the first few epochs, while the CRB continues to decrease. This behavior suggests that the unfolded architecture first learns feasible precoding strategies and then progressively refines them to enhance sensing performance.

The learned penalty weights, shown in Fig. 9a, exhibit distinct behaviors for the power and SINR constraints. The power penalty $\lambda_{\text{p}}^{(t)}$ remains relatively small in the early layers, allowing broader exploration of the solution space, and then increases sharply in the final layers ($t = 8$ to 10) to enforce the power budget more strictly at the network output (Bertsekas, 1999; Boyd; Vandenberghe, 2004). By contrast, the SINR penalty $\lambda_{\text{s}}^{(t)}$ follows a more non-monotonic pattern, increasing during the first few layers, relaxing in the intermediate layers, and rising again near the end. This behavior suggests that the network learns to modulate constraint enforcement differently for the sensing and communication requirements across the unfolded trajectory.

A similar layer-dependent trend is observed for the learned update coefficients. As shown in Fig. 9b, the network adopts relatively large outer step sizes $\beta^{(t)}$ in the initial layers, enabling rapid movement across the optimization landscape during the coarse search phase. In deeper layers, these coefficients decrease, which helps avoid oscillations and supports finer convergence near the final solution. Likewise, the learned inner step sizes in Fig. 9c are generally larger in the early stages and become progressively smaller in later layers. This schedule is consistent with the role of the inner NAGD solver: large early steps favor fast descent, whereas smaller late-stage steps promote stable refinement of the final precoder.



(a) Total training loss

(b) CRB (dB)

(c) Minimum SINR (dB)

(d) Total transmit power

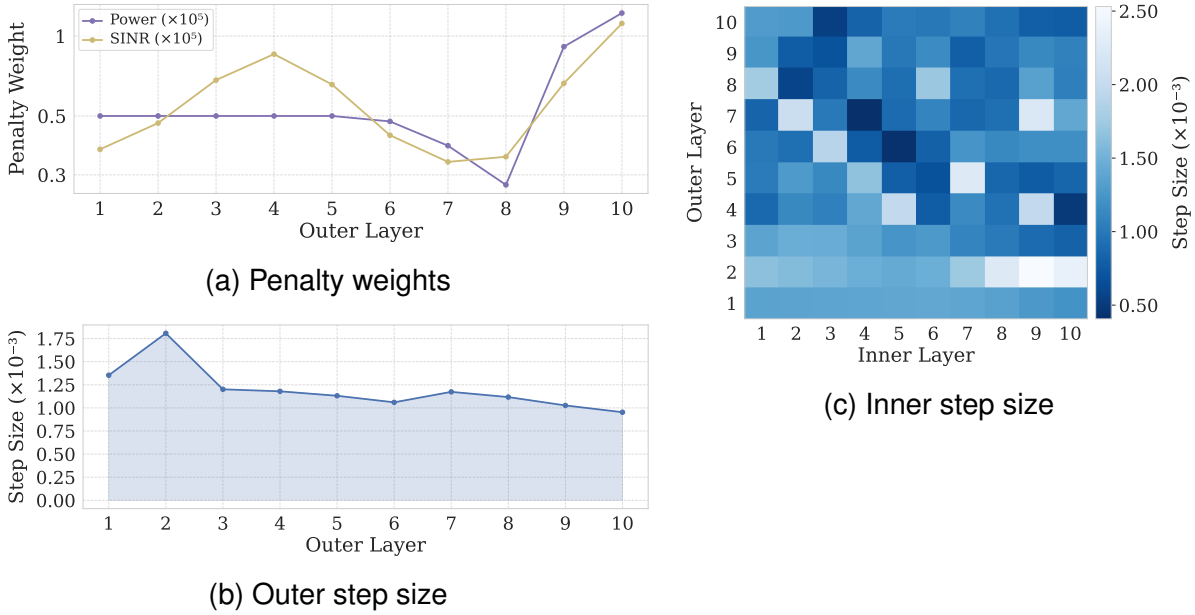
Figure 8 – Training dynamics of the proposed DU-SCA model for $N_t = 16$, $K = 10$, $\Gamma = 15\,\text{dB}$, and $P_T = 1.0\,\text{W}$. The close agreement between training and validation curves indicates good generalization, while the rapid decay of the violation terms shows that the network quickly learns to satisfy the imposed constraints.

3.6.3 Performance Comparison: Baseline Scenario

We compare the proposed DU-SCA framework against the classical iterative SCA algorithm under the baseline scenario.

3.6.3.1 Overall Performance Summary

Table 4 and Fig. 10 summarize the overall performance statistics. The most immediate observation concerns feasibility: the proposed DU-SCA method achieves



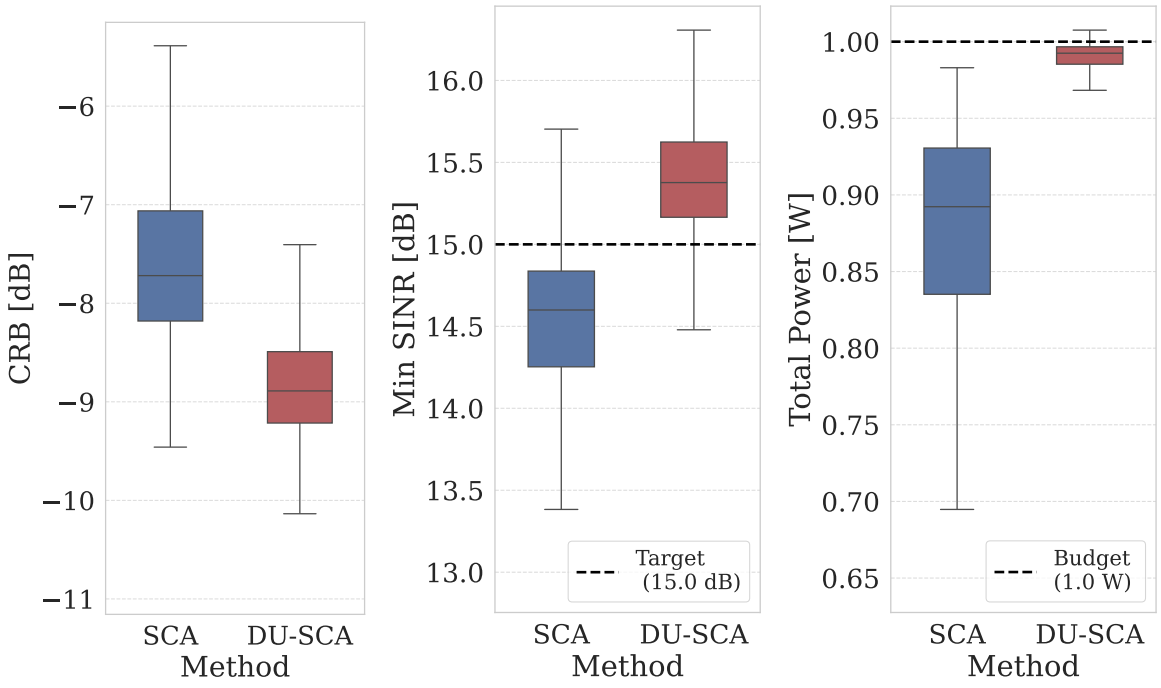
(a) Penalty weights

(b) Outer step size

(c) Inner step size

Figure 9 – Layer-wise evolution of the learnable parameters in the proposed DU-SCA architecture: (a) penalty weights $\lambda^{(t)}$, (b) outer update coefficient $\beta^{(t)}$, and (c) inner step sizes $\eta^{(t,l)}$.

an SINR satisfaction rate of 89.5%, whereas the classical SCA baseline satisfies the communication constraint in only 11.3% of the tested channel realizations. This large gap indicates a substantial improvement in reliability under the same computational budget.



(a) CRB

(b) Minimum SINR

(c) Transmit power

Figure 10 – Statistical performance comparison between the classical SCA solver and the proposed DU-SCA: (a) CRB, (b) minimum SINR, and (c) transmit power.

This improvement in feasibility is closely related to the power-allocation behavior shown in Fig. 10c. The proposed DU-SCA framework makes more effective use of the available transmit power budget, operating much closer to the maximum limit than the classical solver (0.989 W versus 0.877 W), while still maintaining full power-constraint satisfaction. This more efficient resource utilization translates into gains in both communication and sensing performance. In particular, Fig. 10b shows that the proposed method achieves a higher average minimum SINR, while Fig. 10a shows a $\mathbf{1.25}$ dB reduction in CRB, corresponding to improved sensing accuracy.

Table 4 – Constraint Satisfaction and Performance Statistics ($N_t = 16$, $K = 10$, $\Gamma = 15$ dB, $P_T = 1.0$ W)

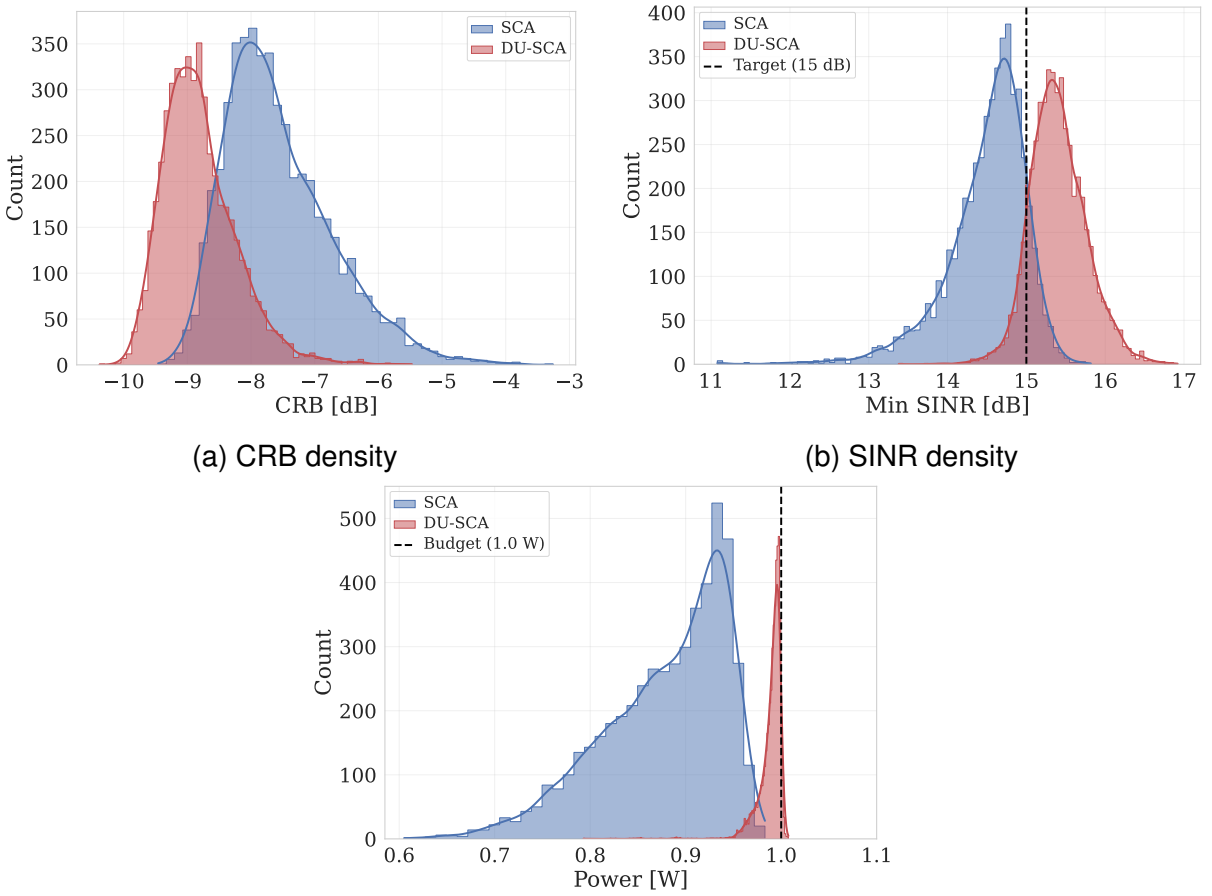
	Satisfaction (%)		**Avg. Metrics (Mean $\pm$ Std)**		
Method	**SINR**	**Power**	**CRB (dB)**	**SINR (dB)**	**Power (W)**
SCA	11.3	100.0	-7.56 ± 0.86	14.49 ± 0.52	0.877 ± 0.065
DU-SCA	**89.5**	**100.0**	$\mathbf{-8.81 \pm 0.58}$	$\mathbf{15.40 \pm 0.36}$	$\mathbf{0.989 \pm 0.013}$

3.6.3.2 Statistical Reliability and Constraint Satisfaction

To further assess the reliability of the proposed method, we examine the empirical distributions of the main performance metrics over the test set. The SINR histogram in Fig. 11b provides a clear view of the feasibility behavior. In particular, the distribution associated with the classical SCA solver is centered below the target threshold of 15 dB, indicated by the dashed green line, which explains its low SINR satisfaction rate. By contrast, the distribution of the proposed DU-SCA method is shifted to the right, with most of its probability mass lying above the target threshold. This confirms that the learned unfolded solver is substantially more reliable in meeting the communication requirement across channel realizations.

The power histogram in Fig. 11c provides additional insight into the convergence behavior of the two methods. The proposed DU-SCA architecture concentrates tightly around the maximum power budget of 1.0 W, indicating that it consistently drives the solution toward the high-power operating point that is favorable for sensing performance. In contrast, the classical SCA solver exhibits a broader distribution over lower power values, suggesting that, under the same computational budget of 100 iterations, it often fails to reach the boundary of the feasible power region. This observation supports the conclusion that the unfolded architecture more effectively exploits the available power budget within the prescribed latency-constrained optimization window.

This reliability gap is further quantified by the Cumulative Distribution Function (CDF) of the achieved minimum SINR, shown in Fig. 12. The vertical line at 15 dB marks the feasibility threshold. The intersection of this threshold with the classical SCA



(a) CRB density

(b) SINR density

(c) Power density

Figure 11 – Empirical distributions of the main performance metrics: (a) CRB, (b) minimum SINR, and (c) transmit power.

curve indicates that only a small fraction of realizations satisfy the target, consistent with the 11.3% feasibility rate reported in Table 4. By contrast, the DU-SCA curve lies markedly to the right, confirming a satisfaction rate of 89.5%. These results reinforce that the proposed unfolded solver is significantly more robust to channel variability than the classical iterative baseline.

3.6.3.3 Convergence Behavior

Fig. 13 illustrates the evolution of the main performance metrics over the optimization process. The most striking difference appears in the convergence of the minimum SINR, shown in Fig. 13b. The classical SCA solver exhibits relatively slow progress and eventually plateaus around 14.6 dB, remaining below the target threshold even after the full budget of 100 iterations. By contrast, the proposed DU-SCA architecture achieves a much faster rise in the early stages of the optimization. By iteration 3, the proposed method already achieves a minimum SINR of approximately 7.3 dB, compared with about -2.5 dB for the classical baseline. By layer 4, the unfolded network reaches approximately 12.1 dB, whereas the classical solver only attains a compara-

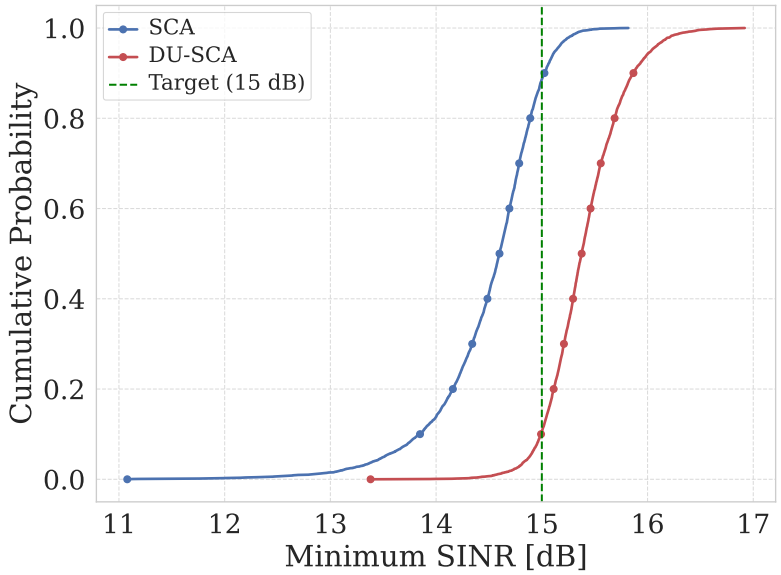
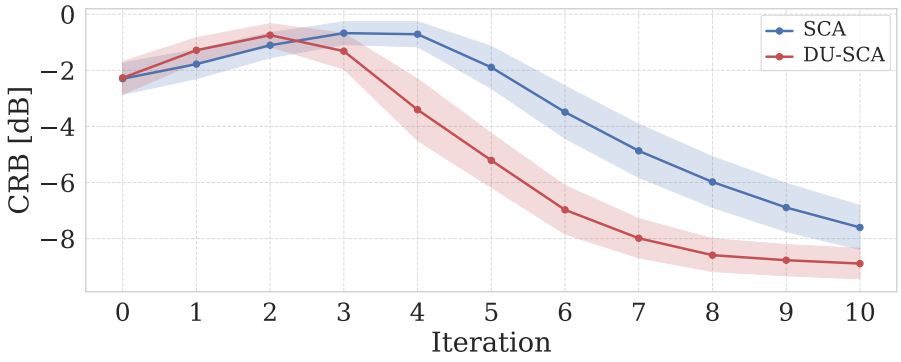


Figure 12 – CDF of the achieved minimum SINR. The vertical line at 15 dB indicates the target threshold, and the intersections with the two curves highlight the large improvement in feasibility rate achieved by the proposed DU-SCA method compared with the classical SCA baseline.

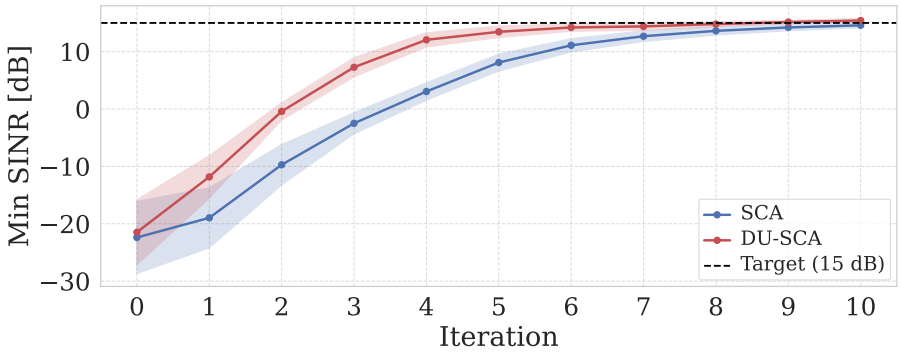
ble level around iteration 7. This early acceleration indicates that the learned update rules effectively mitigate the initial stagnation observed in the baseline method. Ultimately, the proposed DU-SCA framework surpasses the strict 15 dB target by iteration 9, thereby achieving feasibility within the prescribed computational budget.

A similar trend is observed in the sensing objective. As shown in Fig. 13a, once feasibility is approached, the proposed method continues to reduce the CRB more effectively than the classical baseline, converging to a lower final value of approximately -8.8 dB compared with -7.6 dB for the classical SCA solver. Moreover, from the early stages of the optimization onward, the unfolded architecture maintains a consistent advantage in CRB, indicating that the acceleration achieved by the learned updates does not come at the expense of sensing performance.

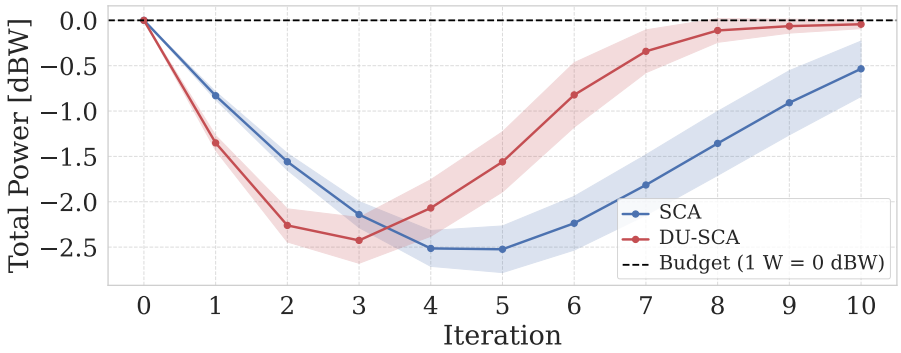
The convergence of the transmit power, shown in Fig. 13c, further clarifies the difference between the two approaches. The classical solver approaches the power limit slowly and often remains below the available budget within the allowed number of iterations. In contrast, the proposed DU-SCA architecture rapidly drives the solution toward the power boundary, recognizing that improved sensing performance is achieved when the available transmit power is more fully exploited. In practice, the unfolded solver saturates the power constraint around iteration 9, which is consistent with its faster attainment of both feasibility and lower CRB.



(a) CRB convergence



(b) SINR convergence



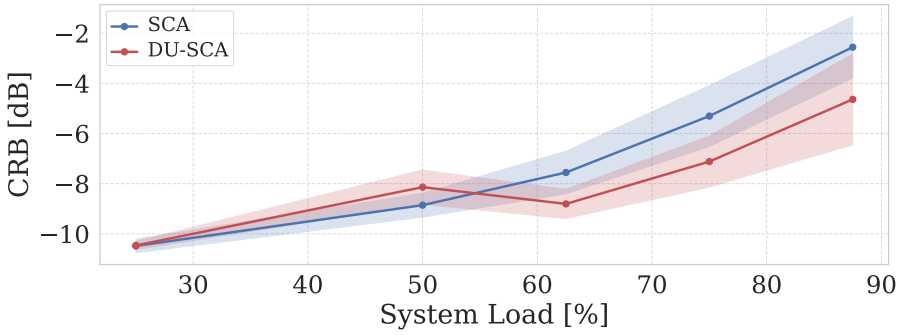
(c) Power convergence

Figure 13 – Evolution of the main performance metrics over the optimization iterations: (a) CRB, (b) minimum SINR, and (c) transmit power. The shaded regions indicate the 95% confidence intervals.

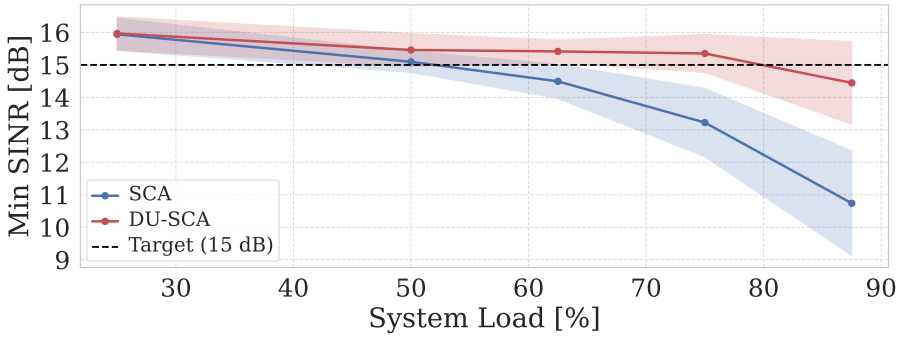
3.6.4 Scalability and Loading Analysis

To evaluate the robustness of the proposed framework under different operating conditions, we vary the number of users from $K = 4$ (light load, 25%) to $K = 14$ (heavily loaded, 87.5%). The resulting performance trends are shown in Fig. 14, where the shaded regions represent the 95% confidence intervals. Three distinct operating regimes can be identified.

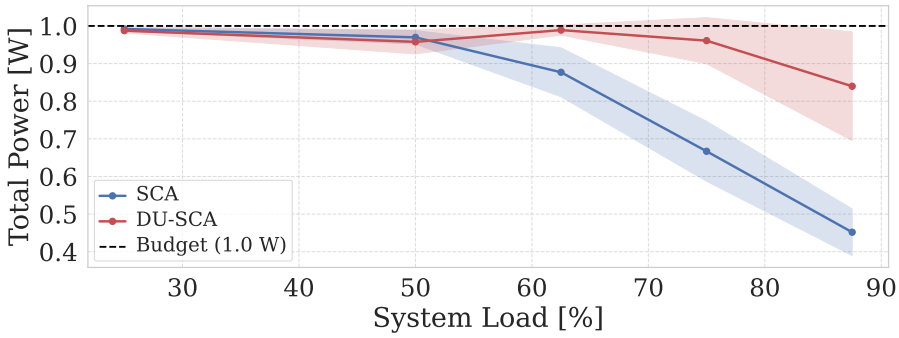
In the low-load regime ($K = 4$ to 8), the system has abundant spatial degrees of freedom, since $K \ll N_t$. As shown in Fig. 14b, the classical SCA solver is already able to satisfy the 15 dB target reliably, and the proposed DU-SCA framework achieves com-



(a) CRB versus user loading



(b) Minimum SINR versus user loading



(c) Transmit power versus user loading

Figure 14 – Performance sensitivity to user loading K: (a) average CRB, (b) minimum SINR, and (c) transmit power. Shaded regions indicate the 95% confidence intervals.

parable performance. In this regime, the classical solver attains a slightly lower CRB at $K = 8$. This behavior is consistent with the fact that the exact iterative solver can more tightly approach the active constraint boundary, whereas the unfolded network, trained for robust performance across a wide range of operating conditions, tends to preserve a small feasibility margin. This slight loss in lightly loaded scenarios is acceptable in practice, since the main objective is robust performance in the more demanding regimes.

A clear performance gap emerges in the transition regime around $K = 10$, corresponding to a loading factor of approximately 62.5%. At this point, the classical SCA solver begins to struggle within the fixed budget of 100 iterations. As shown in Fig. 14c,

it stops short of the power boundary, with an average transmit power of 0.877 W, and fails to meet the SINR target, reaching only 14.49 dB on average. By contrast, the proposed DU-SCA framework uses its learned update rules to reach a more favorable operating point within the same computational budget, pushing the solution close to the power limit (0.989 W), satisfying the SINR requirement (15.40 dB), and achieving a lower CRB of -8.81 dB compared with -7.56 dB for the classical baseline.

In the high-load regime ($K = 12$ to 14), interference management becomes the dominant limitation. The classical solver experiences a marked feasibility degradation, with the average minimum SINR dropping to 10.74 dB and the average transmit power falling to 0.45 W at $K = 14$. This large amount of unused power indicates that, under the imposed latency budget, the baseline method is unable to reach a sufficiently good feasible region. The proposed DU-SCA framework is significantly more robust in this regime, maintaining higher power utilization and substantially better SINR values. Nevertheless, even the unfolded solver cannot fully overcome the fundamental difficulty of the overloaded channel: at $K = 14$, it still exhibits a small SINR shortfall, reaching 14.45 dB instead of the 15 dB target. This increased focus on communication feasibility also comes at the expense of sensing quality, as reflected by the degradation of the CRB to -4.64 dB.

Overall, these results show that the proposed DU-SCA framework is substantially more robust to increasing multi-user interference than the classical gradient-based baseline, particularly in the medium- and high-load regimes where fast and reliable convergence is most critical.

Table 5 – System performance versus user load ($N_t = 16$, $\Gamma = 15$ dB, $P_T = 1.0$ W)

Users K (Load)	**Method**	**CRB (dB)** Mean $\pm$ Std	**Min. SINR (dB)** Mean $\pm$ Std	**Power (W)** Mean $\pm$ Std
4 (25%)	SCA	-10.48 ± 0.27	15.94 ± 0.50	0.99 ± 0.01
	DU-SCA	$\mathbf{-10.47 \pm 0.16}$	$\mathbf{15.97 \pm 0.51}$	$\mathbf{0.99 \pm 0.01}$
8 (50%)	SCA	-8.86 ± 0.48	15.10 ± 0.33	0.97 ± 0.02
	DU-SCA	$\mathbf{-8.14 \pm 0.68}$	$\mathbf{15.46 \pm 0.51}$	$\mathbf{0.96 \pm 0.03}$
10 (62%)	SCA	-7.56 ± 0.85	14.49 ± 0.53	0.88 ± 0.06
	DU-SCA	$\mathbf{-8.81 \pm 0.58}$	$\mathbf{15.42 \pm 0.35}$	$\mathbf{0.99 \pm 0.01}$
12 (75%)	SCA	-5.30 ± 1.22	13.22 ± 1.06	0.67 ± 0.08
	DU-SCA	$\mathbf{-7.12 \pm 1.02}$	$\mathbf{15.35 \pm 0.58}$	$\mathbf{0.96 \pm 0.06}$
14 (88%)	SCA	-2.56 ± 1.23	10.74 ± 1.61	0.45 ± 0.06
	DU-SCA	$\mathbf{-4.64 \pm 1.81}$	$\mathbf{14.45 \pm 1.27}$	$\mathbf{0.84 \pm 0.14}$

3.6.5 Tradeoff Analysis

To characterize the trade-off between sensing accuracy and communication reliability, we sweep the target SINR requirement Γ from 5 dB to 25 dB and evaluate

the corresponding sensing performance and achieved communication quality. Fig. 15 summarizes the resulting trade-off behavior.

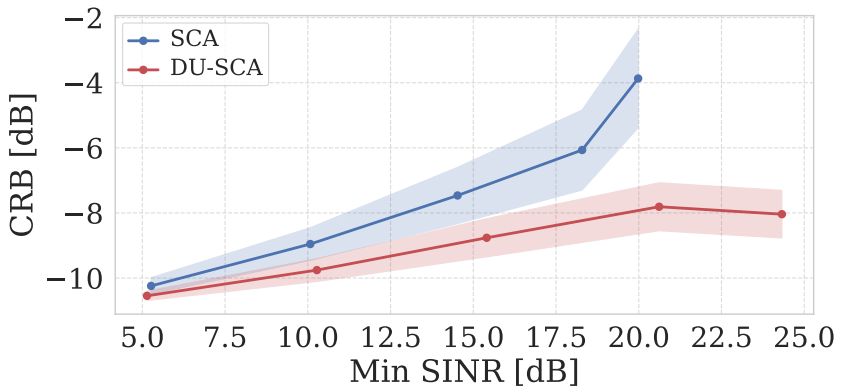


Figure 15 – Trade-off comparison between sensing accuracy (CRB) and communication QoS (SINR).

The results show that the proposed DU-SCA framework consistently outperforms the classical SCA baseline across the tested operating range, yielding a more favorable sensing-communication trade-off. Two main regimes can be identified.

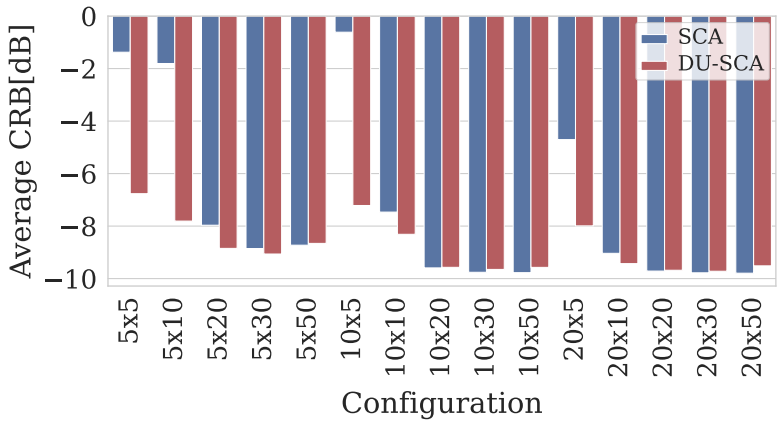
Relaxed-Constraint Regime: In this regime, corresponding to $\Gamma \leq 10$ dB, both methods are able to identify feasible solutions. Even under these relatively loose communication requirements, however, the proposed DU-SCA method retains a measurable advantage, achieving a lower CRB than the classical solver. For example, at $\Gamma = 10$ dB, the sensing gain is approximately 0.8 dB in favor of the unfolded approach.

Stringent-Constraint Regime: When the communication requirement becomes more demanding, corresponding to $\Gamma \geq 15$ dB, the gap between the two methods increases substantially. At $\Gamma = 15$ dB, the classical solver falls slightly below the target, with a deficit of about 0.47 dB, whereas the proposed DU-SCA framework satisfies the requirement (15.40 dB) while simultaneously reducing the CRB by about 1.3 dB. A similar pattern is observed at $\Gamma = 20$ dB, where the classical method exhibits a larger feasibility shortfall of nearly 1.7 dB, while the unfolded solver remains feasible (20.62 dB) and achieves an additional sensing gain of about 1.74 dB.

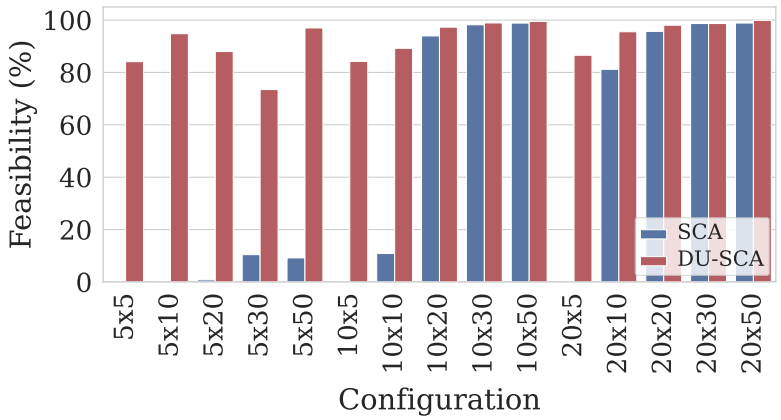
At the most demanding point, $\Gamma = 25$ dB, the system operates close to its practical feasibility limit under the given configuration. In this case, the classical solver suffers a severe performance degradation, falling short of the target by about 5.02 dB. The proposed DU-SCA method remains considerably more robust, missing the target by only 0.67 dB. This result suggests that the unfolded architecture is significantly more effective at navigating the non-convex design space within a limited iteration budget, particularly in regimes where communication constraints are tight and feasible solutions become difficult to reach.

3.6.6 Computational Complexity and Efficiency

To assess the practical efficiency of the proposed framework, we analyze its performance as a function of the total computational budget, defined as $T \times L$, where T denotes the number of unfolded outer layers and L the number of inner optimization steps per layer. Specifically, we vary the depth $T \in \{5, 10, 20\}$ and the number of inner steps $L \in \{5, 10, 20, 30, 50\}$, and report the resulting performance trends against the total number of gradient evaluations in Fig. 16.



(a) Average CRB



(b) Constraint satisfaction rate

Figure 16 – Performance versus computational budget, measured as **outer** $\times$ **inner** iterations: (a) average CRB and (b) constraint satisfaction rate.

The results reveal a clear efficiency advantage for the proposed DU-SCA architecture. Even with a relatively small computational budget, the unfolded framework is able to recover highly feasible solutions. For example, with only $5 \times 10 = 50$ total iterations, the proposed method already achieves a constraint satisfaction rate of 94.8%. Under the same budget, the classical SCA solver fails to produce feasible solutions, yielding a 0.0% satisfaction rate, and requires approximately four times more iterations (200) to attain a comparable level of reliability (93.9%).

In addition, the unfolded architecture exhibits graceful performance degradation

as the computational budget is reduced. Even in the extremely low-budget regime of $5 \times 5 = 25$ total iterations, the proposed DU-SCA framework still maintains a relatively high feasibility rate of 84.1% and an average transmit power of 0.92 W. By contrast, the classical iterative solver deteriorates sharply under the same conditions, indicating that the learned update mechanism is substantially more effective at exploiting limited computational resources.

3.7 Conclusion

This work investigated the trade-off in real-time ISAC systems between the high computational cost of non-convex beamforming optimization and the stringent latency requirements of next-generation wireless networks. To address this challenge, we proposed a model-driven deep unfolding (DU) architecture that reformulates the classical successive convex approximation (SCA) algorithm as a fixed-depth computational graph. In this framework, static heuristic hyperparameters are replaced by learnable update schedules, enabling adaptive and data-driven optimization while preserving the structure of the original model-based solver.

Extensive numerical results showed that the main limitation of conventional iterative solvers in high-interference and latency-constrained settings is not necessarily poor asymptotic performance, but rather their inability to reach high-quality feasible solutions within a restricted computational budget. In the baseline loaded scenario ($K = 10$), the classical solver achieved feasible solutions in only 11.3% of the tested channel realizations, whereas the proposed DU-SCA framework reached an 89.5% feasibility rate under the same iteration budget. This acceleration enabled more effective use of the available transmit power and translated into sensing gains on the order of 1.25 to 1.75 dB in CRB, depending on the operating point.

The results also highlight the limits of the proposed approach. In highly loaded regimes, where the available spatial degrees of freedom become scarce, the unfolded architecture naturally shifts its emphasis toward maintaining communication feasibility, which leads to a corresponding degradation in sensing performance. Moreover, although the learned optimizer significantly enlarges the practically achievable operating region, it remains fundamentally constrained by the feasibility limits of the underlying system configuration, as evidenced by the residual violations observed at very demanding SINR targets such as $\Gamma = 25$ dB.

A further limitation is that the learned update schedules are tied to the operating distribution used during training. Parameters such as the number of users, the user loading factor, the channel statistics, the SINR target, and the transmit-power budget directly affect the geometry of the feasible set and the relative difficulty of the

sensing–communication trade-off. Therefore, when the deployment scenario changes substantially, for example when the system moves from a moderate-load regime to a heavily loaded one, the trained DU-SCA model may no longer provide the best precision or feasibility performance. In such cases, retraining or at least fine-tuning on data representative of the new scenario is required to recover the best operating point. This dependence on scenario-specific training is an important practical cost of the proposed method and should be considered when the environment varies rapidly or when many operating configurations must be supported.

From a computational perspective, the proposed framework is attractive because it transfers most of the adaptation effort to an offline training stage and enables fast online inference. However, this benefit comes with a trade-off: if the unfolded architecture must become excessively deep to handle more stringent scenarios, the training burden may become substantial. In such cases, especially when the channel statistics vary slowly, directly tuning the hyperparameters of a classical SCA solver through offline search may remain a competitive alternative.

Overall, the proposed DU-SCA framework provides a practical and interpretable foundation for low-latency beamforming in future ISAC systems. Future work will investigate robust precoding under CSI uncertainty and extend the proposed paradigm to wideband and millimeter-wave settings.

4 MITIGATING NON-LINEAR DISTORTIONS IN PRACTICAL IMPLEMENTATION OF IEEE 802.15.7 CSK MODULATION USING MACHINE LEARNING

4.1 Background and Motivation

Next-generation wireless communication systems are expected to support a broad range of requirements, including EMBB, MMTC, and ubiquitous connectivity (Wang *et al.*, 2023; Singh *et al.*, 2025). Meeting these demands poses substantial challenges to network design, particularly because the continuous growth in the number of connected devices and the increasing demand for high-data-rate services place severe pressure on the available radio-frequency spectrum (Jovicic; Li; Richardson, 2013; Wang *et al.*, 2023; Singh *et al.*, 2025; Saad; Bennis; Chen, 2020). As a result, achieving high spectral efficiency has become a fundamental requirement for future communication systems (Wang *et al.*, 2023; Singh *et al.*, 2025). Although the exploration of higher frequency bands may partially alleviate spectrum scarcity, it also introduces important challenges related to propagation losses, device complexity, and link reliability (Wang *et al.*, 2023; Singh *et al.*, 2025).

In this context, VLC has emerged as a promising complementary wireless technology (Jovicic; Li; Richardson, 2013; Chowdhury *et al.*, 2018). In VLC systems, the visible light spectrum, i.e., the portion of the electromagnetic spectrum perceptible to the human eye, spanning wavelengths from 380 nm to 780 nm, is used as the transmission medium (Dimitrov; Haas, 2015; Geng *et al.*, 2022). The availability of an abundant unlicensed bandwidth exceeding 300 THz makes this spectral range particularly attractive for high-capacity wireless applications (Wang *et al.*, 2017).

A major advantage of VLC lies in its ability to leverage existing solid-state lighting infrastructure for the dual purpose of illumination and data transmission (Jovicic; Li; Richardson, 2013; Pathak *et al.*, 2015; Chowdhury *et al.*, 2018). This characteristic can reduce deployment costs while enabling the integration of communication functionalities into environments where lighting systems are already widely available (Jovicic; Li; Richardson, 2013; Pathak *et al.*, 2015; Chowdhury *et al.*, 2018). In addition, the high modulation speed of light-emitting diodes and the large available bandwidth of the visible spectrum provide the potential for high data rates (Pathak *et al.*, 2015). For these reasons, VLC is widely regarded as a viable solution to complement conventional radio-frequency systems, especially in scenarios where spectrum scarcity, electromagnetic interference, or dense user deployment limit the performance of traditional wireless networks (Jovicic; Li; Richardson, 2013; Chowdhury *et al.*, 2018).

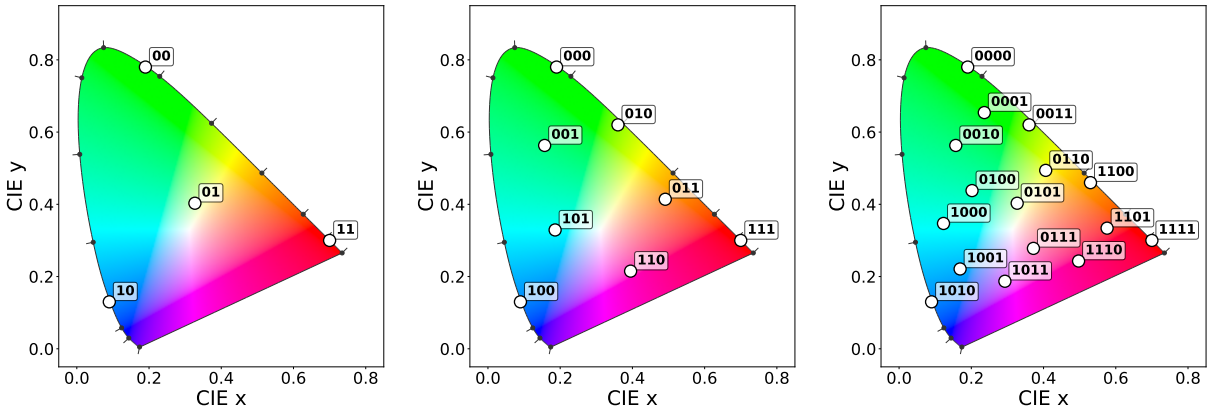
4.1.1 IEEE 802.15.7 Color Shift Keying

Conventional VLC modulation schemes primarily encode data by varying the overall light intensity or by employing discrete on-off keying (Pathak *et al.*, 2015). However, in systems utilizing multicolor light sources, information can instead be transmitted by manipulating the relative optical intensities of distinct color components while strictly preserving the transmitter's perceived illumination and providing higher data rates (Pathak *et al.*, 2015). This mechanism forms the basis of CSK. To standardize this approach, IEEE 802.15.7 specifies CSK within the PHY III operating mode. This physical layer is designed specifically for multi-source, multi-detector optical links, supporting data rates in the tens of megabits per second (IEEE, 2019).

CSK is especially attractive in scenarios where white light is generated by combining independent red, green, and blue light-emitting diodes. In such systems, data transmission can be achieved by jointly modulating the optical intensities of the three emitters. This approach avoids the bandwidth limitations associated with phosphor-converted white Light-Emitting Diodes (LEDs) and enables the exploitation of multicolor sources for both illumination and communication (Pathak *et al.*, 2015). In this context, the transmitted symbols are represented as chromaticity points, so that information is encoded in color variations rather than in the intensity of a single optical carrier.

According to IEEE 802.15.7, the CSK signal is generated using three color light sources selected from the seven optical bands defined in the standard. These three sources establish the vertices of a constellation triangle in the CIE 1931 chromaticity diagram, and each transmitted symbol corresponds to a point inside this triangle (Pathak *et al.*, 2015; IEEE, 2019). Depending on the modulation order, the standard defines 4-CSK, 8-CSK, and 16-CSK constellations, which respectively map 2, 3, and 4 bits per symbol, as illustrated in Fig. 17 (Pathak *et al.*, 2015; IEEE, 2019). The constellation design must simultaneously satisfy communication and illumination requirements: on the one hand, the symbols should be sufficiently separated in the chromaticity plane to improve detection robustness; on the other hand, they should remain distributed in a way that preserves the desired perceived color of the emitted light (Pathak *et al.*, 2015; IEEE, 2019).

After the chromaticity coordinates associated with a symbol are determined, the corresponding optical powers of the three emitters are computed from the color-mixing relations imposed by the selected RGB triangle. Therefore, transmission is ultimately performed by adjusting the relative intensities of the three optical channels so that the emitted light reproduces the target chromaticity of the symbol (Pathak *et al.*, 2015; IEEE, 2019). At the receiver, the optical components associated with the three colors are measured and processed to estimate the transmitted symbol, which requires proper color mapping and receiver calibration.



(a) 4-CSK (b) 8-CSK (c) 16-CSK

Figure 17 – IEEE 802.15.7 CSK constellations mapped onto the CIE 1931 chromaticity diagram: (a) 4-CSK, (b) 8-CSK, and (c) 16-CSK.

Although the IEEE 802.15.7 CSK specification relies on an idealized chromatic representation of transmitted symbols (IEEE, 2019), practical VLC systems utilizing commercial LEDs are limited by physical nonidealities. Unlike the standard's assumption of monochromatic light sources, real LEDs are characterized by finite spectral linewidths and inherent spectral overlap (Mathias; de Melo; Abrao, 2021). Together with receiver-side imperfections, these hardware constraints disrupt the intended color-mixing process, yielding significant inter-channel crosstalk and structural constellation deformation. This translates into a highly nonlinear mapping between transmitted symbols and received signals, fundamentally invalidating the use of ideal decision regions for symbol recovery (Mathias; de Melo; Abrao, 2021). Consequently, robust detection in practical CSK systems requires a carrefull joint modeling approach that simultaneously accounts for nonlinear distortion and receiver noise.

4.2 System Model

This section develops the practical IEEE 802.15.7 CSK link model used throughout the chapter. The presentation follows the physical transceiver chain: symbol synthesis at the RGB transmitter, propagation through a coupled optical-electrical channel, linear equalization and chromaticity reconstruction at the receiver, and finally the noise abstraction adopted in the Monte Carlo datasets. This ordering is important because the observed distortion does not originate in a single block; it emerges from the combined effect of non-ideal emitters, non-ideal receiver filtering, and the linear operations used for symbol recovery.

Fig. 18 summarizes this processing chain. Input bits are first mapped onto nominal chromaticity points in the CIE 1931 plane. These points are then translated into practical RGB drive levels compatible with the measured spectra of the adopted

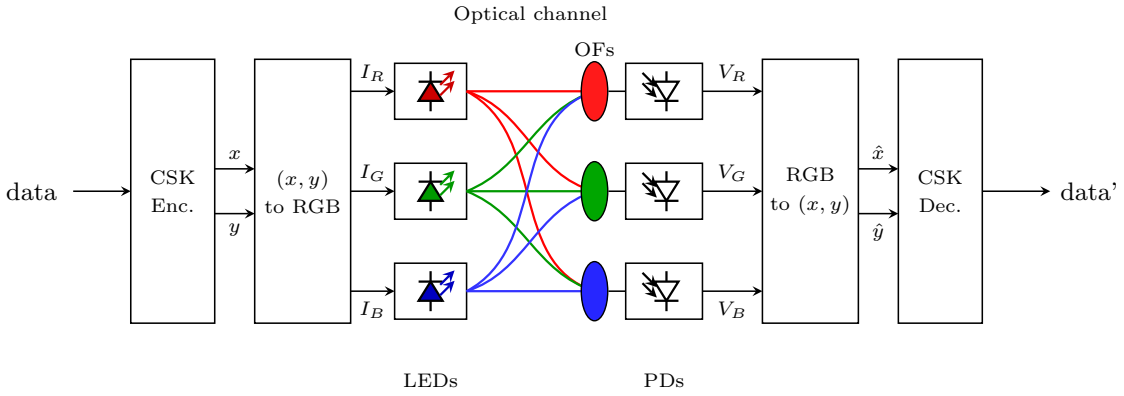


Figure 18 – Block diagram of the considered IEEE 802.15.7 CSK transceiver, highlighting symbol generation, optical propagation, receiver equalization, and chromaticity-based symbol recovery.

LEDs. After optical propagation and photodetection, the receiver observes a three-dimensional voltage vector affected by crosstalk and noise. A linear equalization stage reduces channel coupling, after which the equalized RGB estimate is projected back onto the chromaticity plane for symbol decision. Throughout this chapter, $[\cdot]^{\mathsf{T}}$ denotes transpose.

4.2.1 Transmitter-Side Chromaticity Synthesis

In IEEE 802.15.7 CSK, each symbol is specified by a target chromaticity point in the CIE 1931 plane. A practical RGB transmitter, however, does not choose chromaticity directly; it only controls the relative drive levels of the red, green, and blue emitters. The first task is therefore to solve the following inverse problem: for a given reference symbol, which RGB mixture produces the closest achievable chromaticity using the adopted Off the Shelf (OTS) LEDs.

Addressing this requires first defining the forward mapping from a candidate emission spectrum to the CIE 1931 color space. Specifically, for an arbitrary optical spectrum $S(\lambda)$, the corresponding tristimulus vector is given by (Wang *et al.*, 2017):

$$\mathbf{C} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \int_{\lambda_{\mathsf{L}}}^{\lambda_{\mathsf{H}}} S(\lambda)\, \bar{x}(\lambda)\, d\lambda \\ \int_{\lambda_{\mathsf{L}}}^{\lambda_{\mathsf{H}}} S(\lambda)\, \bar{y}(\lambda)\, d\lambda \\ \int_{\lambda_{\mathsf{L}}}^{\lambda_{\mathsf{H}}} S(\lambda)\, \bar{z}(\lambda)\, d\lambda \end{bmatrix}, \tag{4.1}$$

where $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are the CIE 1931 color-matching functions over the visible interval $[\lambda_{\mathsf{L}}, \lambda_{\mathsf{H}}]$ presented in the Fig. 19. The corresponding chromaticity coordinates

are then obtained by normalizing the tristimulus values (Wang *et al.*, 2017):

$$\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \dfrac{X}{X+Y+Z} \\ \dfrac{Y}{X+Y+Z} \end{bmatrix}. \tag{4.2}$$

Eqs. (4.1) and (4.2) therefore provide the forward mapping from an emitted spectrum to a point in the chromaticity plane.

Fig. 19 shows the CIE color matching functions together with the normalized spectra of the practical red, green, and blue OTS LEDs used in the implementation. These devices are modeled after the Gentian® RGB sources reported in Mathias, de Melo, and Abrao (2021), with peak wavelengths of 629, 525, and 473 nm, respectively. Fig. 19 shows that the adopted RGB emitters are broadband and partially overlapping. This indicates that practical color mixing departs from the ideal monochromatic source model assumed in the IEEE reference constellation.

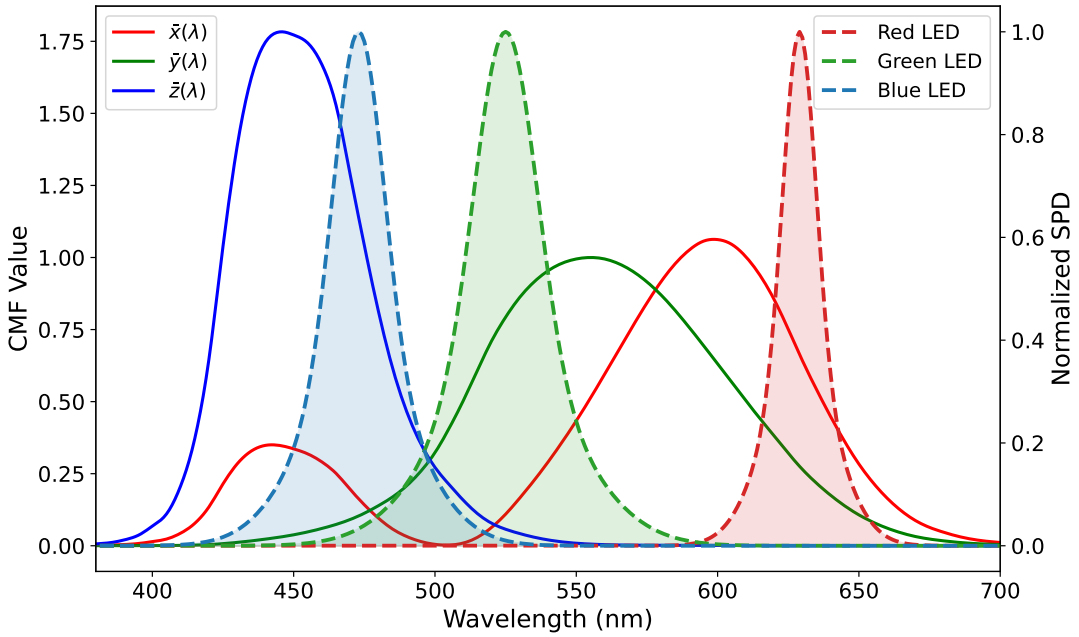


Figure 19 – CIE 1931 color-matching functions and normalized spectra of the adopted RGB OTS LEDs. The practical emitters are broadband and overlap spectrally, which is one of the sources of constellation distortion.

Let $S_{\mathrm{R}}(\lambda)$, $S_{\mathrm{G}}(\lambda)$, and $S_{\mathrm{B}}(\lambda)$ denote the measured spectral distributions of the three emitters. If the three branches are driven with normalized power coefficients collected in the vector $\mathbf{P} = [P_{\mathrm{R}}\, P_{\mathrm{G}}\, P_{\mathrm{B}}]^{\mathsf{T}}$, the emitted spectrum is modeled as (Wang *et al.*, 2017):

$$S_{\mathbf{P}}(\lambda) = P_{\mathrm{R}} S_{\mathrm{R}}(\lambda) + P_{\mathrm{G}} S_{\mathrm{G}}(\lambda) + P_{\mathrm{B}} S_{\mathrm{B}}(\lambda).$$

For each IEEE reference symbol $\mathbf{s}_{\mathrm{ref}}$, the practical RGB drive coefficients are obtained by solving:

$$\mathbf{P}^{*} = \arg \min_{\mathbf{P} \in [0,1]^3} \left\| \mathbf{s}_{\mathrm{ref}} - \mathbf{s}_{\mathbf{P}} \right\|_2 ,$$

where $\mathbf{s_P}$ is the chromaticity produced by $S_{\mathbf{P}}(\lambda)$. This optimization has a simple interpretation: rather than assuming that the practical transmitter can reproduce the nominal IEEE point exactly, it selects the RGB mixture whose chromaticity lies closest to that target. The result is a feasible transmitter constellation described by practical chromaticity points and their associated normalized RGB power coefficients.

Table 6 reports the final operating points used for 4-CSK, 8-CSK and 16-CSK. Bold chromaticity entries indicate symbols for which the practical OTS realization differs from the nominal IEEE location. This mismatch is one of the factors that later degrades minimum-distance detection.

Table 6 – Positions of the symbols in the color space and respective normalized electrical powers of the OTS LEDs

Modulation	Symbol	P_R	P_G	P_B	Achieved $[x, y]$
4-CSK	00	0.0000	1.0000	0.0000	[**0.1674**, **0.7206**]
	01	0.3580	0.4100	0.2320	[**0.2811**, **0.4384**]
	10	0.0000	0.0000	1.0000	[**0.1202**, **0.0912**]
	11	1.0000	0.0000	0.0000	[0.6988, 0.3011]
8-CSK	000	0.0000	1.0000	0.0000	[**0.1674**, **0.7206**]
	001	0.0000	0.8000	0.2000	[**0.1572**, **0.5841**]
	010	0.3170	0.6830	0.0000	[**0.2822**, **0.6300**]
	011	0.6430	0.2950	0.0620	[**0.4329**, **0.4566**]
	100	0.0000	0.0000	1.0000	[**0.1202**, **0.0912**]
	101	0.0860	0.4170	0.4970	[**0.1672**, **0.3629**]
	110	0.5720	0.0340	0.3940	[**0.3637**, **0.1998**]
	111	1.0000	0.0000	0.0000	[0.6988, 0.3011]
16-CSK	0000	0.0000	1.0000	0.0000	[**0.1674**, **0.7206**]
	0001	0.0000	1.0000	0.2720	[**0.1563**, **0.5718**]
	0010	0.0000	1.0000	0.3286	[**0.1545**, **0.5490**]
	0011	0.3570	0.4518	0.0000	[**0.3350**, **0.5883**]
	0100	0.0000	0.4533	0.3299	[**0.1465**, **0.4422**]
	0101	0.6226	0.4194	0.2868	[0.3270, 0.4030]
	0110	1.0000	0.7698	0.0000	[**0.3900**, **0.5449**]
	0111	0.9621	0.1941	0.4818	[0.3720, 0.2780]
	1000	0.0000	0.2435	0.3295	[**0.1393**, **0.3458**]
	1001	0.0000	0.0000	0.3300	[**0.1202**, **0.0912**]
	1010	0.0000	0.0000	1.0000	[**0.1202**, **0.0912**]
	1011	0.5751	0.0000	0.6597	[**0.3036**, **0.1578**]
	1100	1.0000	0.6441	0.0000	[**0.4120**, **0.5275**]
	1101	1.0000	0.1999	0.1431	[**0.4820**, **0.3592**]
	1110	1.0000	0.0000	0.5870	[**0.3870**, **0.1880**]
	1111	1.0000	0.0000	0.0000	[0.6988, 0.3011]

Bold $[x\, y]$ entries indicate OTS-feasible chromaticities that differ from the corresponding IEEE reference symbols by $\Delta > 0.01$.

The result of this first step is not yet the actual channel input. The optimization above determines feasible chromaticities and the corresponding normalized RGB power coefficients, whereas the channel model in Section 4.2.3 is written in terms of LED drive currents. A second transmitter-side step is therefore required to map the normalized coefficients into electrical currents.

4.2.2 Electrical Drive Current Mapping

Once the normalized RGB power coefficients have been determined, they are converted into LED forward currents through the polynomial electrical power–current model reported in Mathias, de Melo, and Abrao (2021), namely:

$$\mathbf{I} = \begin{bmatrix} I_{\mathsf{R}} \\ I_{\mathsf{G}} \\ I_{\mathsf{B}} \end{bmatrix} = \begin{bmatrix} -6.174 \times 10^{-2} P_{\mathsf{R}}^2 + 4.855 \times 10^{-1} P_{\mathsf{R}} - 4.510 \times 10^{-3} \\ -3.471 \times 10^{-2} P_{\mathsf{G}}^2 + 3.586 \times 10^{-1} P_{\mathsf{G}} - 3.476 \times 10^{-3} \\ -3.271 \times 10^{-2} P_{\mathsf{B}}^2 + 3.471 \times 10^{-1} P_{\mathsf{B}} - 3.896 \times 10^{-3} \end{bmatrix}. \tag{4.3}$$

4.2.3 Optical-Electrical Channel Formation

Once the RGB currents are defined, the next step is to describe how they are transformed into receiver voltages. Because the LED spectra overlap and the receiver optical filters are non-ideal, the practical front-end cannot be modeled as three independent color branches. Instead, a coupled three-input three-output model is required (Mathias; Souza; Abrão, 2020; Tong *et al.*, 2024).

The received voltage vector is thus written as (Mathias; Souza; Abrão, 2020):

$$\mathbf{V} = \mathbf{H}\mathbf{I} + \mathbf{n}, \tag{4.4}$$

where $\mathbf{n} = [n_{\mathsf{R}}\, n_{\mathsf{G}}\, n_{\mathsf{B}}]^{\mathsf{T}}$ is a zero-mean AWGN vector whose variance is described in Section 4.2.6, and $\mathbf{H} \in \mathbb{R}^{3\times 3}$ is the channel matrix. The diagonal entries of $\mathbf{H}$ represent the direct gains of the red, green, and blue branches, whereas the off-diagonal entries quantify the residual crosstalk between colors.

Each element of the channel matrix represents the aggregate end-to-end gain from the j-th LED to the i-th photodetector branch. The model combines three physical effects: free-space Lambertian propagation, spectral overlap between the source and the receiver branch, and receiver-side voltage scaling. Accordingly, the (i, j)-th coefficient is given by (Mathias; de Melo; Abrao, 2021; Wang *et al.*, 2017; Dimitrov; Haas, 2015):

$$H_{ij} = \kappa\, \Omega_{ij}\, G_{\mathrm{T}}\, \eta_j\, \Gamma_{ij} \quad [\text{mV/A}], \tag{4.5}$$

where Ω_{ij} is the Lambertian path-loss factor, G_{T} is the effective transimpedance-stage gain used to convert photodiode current into the receiver voltage level adopted by the

channel model, η_j is the efficiency of the j-th LED, Γ_{ij} is the spectral overlap integral, and κ is the maximum photodetector responsivity (Mathias; de Melo; Abrao, 2021).

The Lambertian path-loss factor for line-of-sight propagation is (Wang *et al.*, 2017; Dimitrov; Haas, 2015):

$$\Omega_{ij} = \frac{(m+1)\, A_{\mathrm{pd}}}{2\pi\, d_{ij}^2}\, \cos^{m+1}\phi_{ij}\, \cos\psi_{ij}, \tag{4.6}$$

where m is the Lambertian mode number, A_{pd} is the photodetector active area, d_{ij} is the distance between the j-th LED and the i-th photodetector, ϕ_{ij} is the emission angle, and ψ_{ij} is the incidence angle. The spectral overlap integral between the j-th LED and the i-th receiver branch is (Wang *et al.*, 2017; Dimitrov; Haas, 2015; Mathias; de Melo; Abrao, 2021):

$$\Gamma_{ij} = \int_{\lambda_{\mathrm{L}}}^{\lambda_{\mathrm{H}}} \hat{S}_j(\lambda)\, T_i(\lambda)\, \hat{R}_i(\lambda)\, d\lambda, \tag{4.7}$$

where $\hat{S}_j(\lambda)$ is the peak-normalized LED spectral distribution, $T_i(\lambda)$ is the optical-filter transmittance, and $\hat{R}_i(\lambda)$ is the peak-normalized photodetector responsivity. The non-zero off-diagonal terms in Eq. (4.5) arise precisely because $\Gamma_{ij} \neq 0$ for $i \neq j$, which is a direct consequence of the broadband LED spectra and non-ideal optical filters.

This work adopts the same family of LEDs, Optical Filters (OFs), and photodiodes characterized in Mathias, de Melo, and Abrao (2021). The channel is evaluated on a 380–780 nm, using a 1 m line-of-sight link, 25 mm inter-element spacing, Lambertian mode number $m = 10$, and a channel-model transimpedance scaling of $G_{\mathrm{T}} = 10 \times 10^4$ V/A. The modeled filter transmittances and photodetector responsivity are shown in Fig. 20.

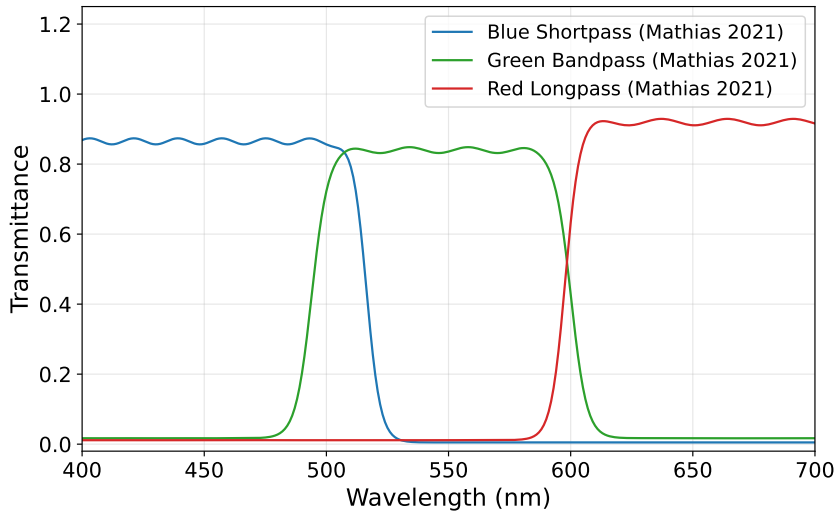
Thus, the resulting channel matrix is:

$$\mathbf{H} = \begin{bmatrix} 132.98 & 2.18 & 1.62 \\ 3.66 & 141.80 & 14.06 \\ 0.72 & 45.32 & 122.77 \end{bmatrix} \quad [\mathrm{mV/A}]. \tag{4.8}$$

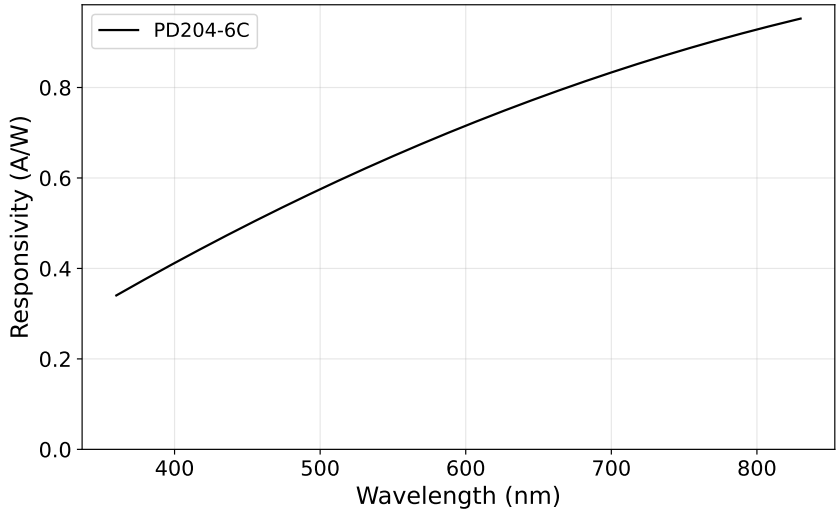
The large off-diagonal terms in the green and blue rows ($H_{23} = 14.06$ and $H_{32} = 45.32$ mV/A) indicate considerable spectral leakage between these two branches. Consequently, the receiver must include an explicit channel-decoupling stage before symbol decision. Once $\mathbf{H}$ is specified, the remaining deterministic receiver operations reduce to equalization and chromaticity reconstruction, which are described next.

4.2.4 Receiver Equalization and Chromaticity Reconstruction

At the receiver, reverse-biased PIN photodiodes convert the incident optical power into electrical current. Each branch produces a photocurrent proportional to the received radiant power through a wavelength-dependent responsivity (Dimitrov; Haas,



(a) Optical-filter transmittances



(b) Photodetector responsivity

Figure 20 – Modeled receiver front-end responses used to build the CSK channel. These responses, together with the LED spectra in Fig. 19, determine the inter-channel spectral coupling.

2015; Wang *et al.*, 2017; Kahn; Barry, 1997). Because the responsivity is non-flat and each branch is preceded by a distinct optical filter, the detector outputs do not form an ideal RGB basis. Instead, each branch measures a weighted spectral mixture of all transmitted components, as reflected by the off-diagonal terms of $\mathbf{H}$.

After photodetection and transimpedance amplification, the receiver applies linear equalization to obtain the RGB estimate:

$$\mathbf{V}_{\mathrm{CE}} = \hat{\mathbf{H}}^{\dagger}\mathbf{V}, \tag{4.9}$$

where $(\cdot)^{\dagger}$ denotes the inverse or, more generally, the Moore–Penrose pseudo-inverse. This step compensates the dominant RGB crosstalk, but it does not guarantee recovery of the nominal IEEE constellation because the transmitter operating points are already

distorted.

To compare the equalized observation with the nominal chromaticity symbols, the receiver maps the equalized RGB vector to the CIE 1931 tristimulus domain through (Fairman; Brill; Hemmendinger, 1997):

$$\mathbf{C}'_{\mathsf{CE}} = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{4.10}$$

after which Eq. (4.2) yields the reconstructed chromaticity estimate $\mathbf{s}'_{\mathsf{CE}} = [x'\, y']^{\mathsf{T}}$. The equalized symbols in Fig. 21 confirm the central point of the chapter: equalization improves class separation, but does not fully restore the ideal geometry. The residual warp is what makes purely geometric detection increasingly suboptimal for practical higher-order CSK.

Fig. 21 is best interpreted in the order imposed by the receiver chain. White circles denote the nominal IEEE symbols, green markers denote the practical transmitter chromaticities obtained from the synthesis step, and blue markers denote the equalized received points after channel inversion. Read in that order, the figure makes the logic of the chapter explicit: the first mismatch appears at symbol synthesis, and the remaining mismatch after equalization is exactly what motivates the learning-based detectors introduced later.

The deterministic part of the end-to-end model is therefore complete at this stage: a transmitted symbol is mapped to a practical RGB current vector, propagated through the coupled channel, equalized, and finally reconstructed as a chromaticity estimate.

As a synthesis of the model developed so far, for a transmitted symbol index m, the end-to-end chain considered in this chapter can be summarized as

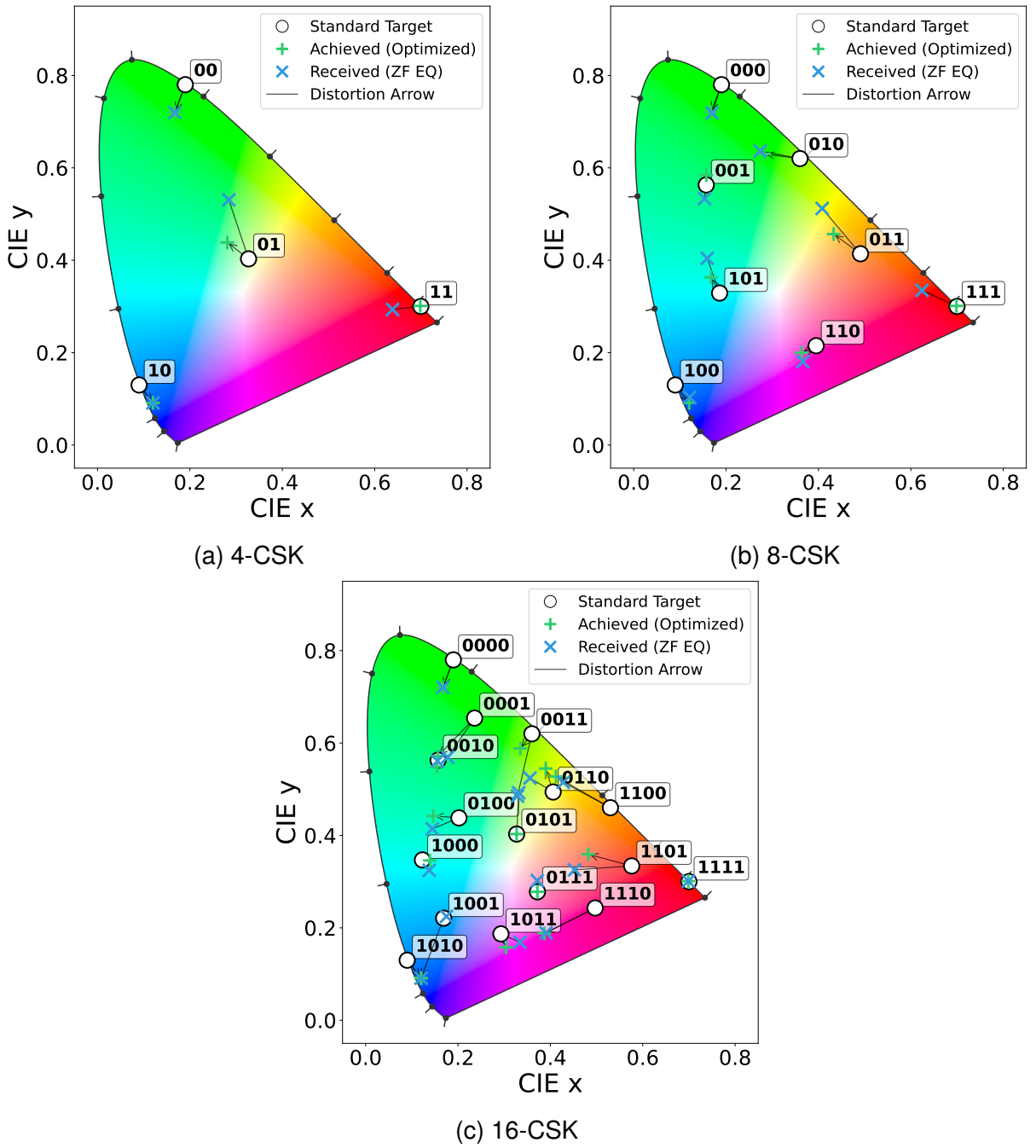
$$m \;\longrightarrow\; \mathbf{s}_m^{\mathrm{ref}} \;\longrightarrow\; \mathbf{P}_m \;\longrightarrow\; \mathbf{I}_m \;\longrightarrow\; \mathbf{V} = \mathbf{H}\mathbf{I}_m + \mathbf{n} \;\longrightarrow\; \mathbf{V}_{\mathsf{CE}} \;\longrightarrow\; \mathbf{s}'_{\mathsf{CE}}.$$

4.2.5 Symbol Decision by Minimum Euclidean Distance

Once the reconstructed chromaticity estimate $\mathbf{s}'_{\mathsf{CE}}$ is available, the receiver must decide which constellation symbol was transmitted. The conventional approach is MED: each observation is assigned to the symbol whose reference chromaticity is closest in the Euclidean sense:

$$\hat{m} = \arg \min_{m \in \{1,\ldots,M\}} \left\| \mathbf{s}'_{\mathsf{CE}} - \mathbf{s}_m^{\mathrm{ref}} \right\|_2, \tag{4.11}$$

where $\mathbf{s}_m^{\mathrm{ref}}$ is the reference chromaticity of the m-th constellation symbol and M is the constellation order. This rule requires no training data and has minimal computational cost, which makes it the natural default for symbol recovery. Its decision boundaries,



(a) 4-CSK

(b) 8-CSK

(c) 16-CSK

Figure 21 – End-to-end constellation evolution in the CIE 1931 plane. White circles denote IEEE reference symbols, green markers denote the chromaticities generated by the adopted OTS RGB sources, and blue markers denote the equalized received points.

however, are fixed Voronoi regions determined by the nominal constellation rather than by the practical received clusters. As Fig. 21 illustrates, equalization does not fully restore the ideal symbol positions, so these fixed regions can become increasingly suboptimal as the constellation order grows and the residual warp becomes more pronounced.

The deterministic receiver chain, including the symbol decision rule, is therefore complete at this stage. The only remaining ingredient is the stochastic perturbation introduced by the receiver front-end, which is specified next.

4.2.6 Noise Model

The deterministic chain above explains how constellation warping is created. To complete the model, the receiver noise must also be specified. The treatment adopted here proceeds at two levels. First, the physical noise terms are written in input-referred current form. Second, those terms are collapsed into a single voltage-domain parameter σ_{n} for dataset generation and detector comparison.

For branch $i \in \{\mathtt{R}, \mathtt{G}, \mathtt{B}\}$, let $i_{\mathrm{sig},i}(t)$ denote the useful photocurrent and let $i_{\mathrm{n},i}(t)$ denote the aggregate input-referred noise current entering the transimpedance amplifier (TIA). The branch voltage is then modeled as:

$$v_i(t) = G_{\mathrm{T}} \left(i_{\mathrm{sig},i}(t) + i_{\mathrm{n},i}(t) \right), \tag{4.12}$$

where $G_{\mathrm{T}} = 10^4$ V/A is the reference Transimpedance Amplifier (TIA) gain.

The total input-referred noise variance for each branch is the sum of contributions from shot and thermal noises (Kahn; Barry, 1997):

$$\sigma^2_{\mathrm{total},i} = \sigma^2_{\mathrm{shot},i} + \sigma^2_{\mathrm{thermal},i}. \tag{4.13}$$

Shot noise is induced primarily by the received ambient optical power $P_{n,i}$, which causes a steady DC photocurrent (Kahn; Barry, 1997). The shot-noise variance is modeled as essentially white and Gaussian, and is given by (Kahn; Barry, 1997; Wang *et al.*, 2017):

$$\sigma^2_{\mathrm{shot},i} = 2qR_i P_{n,i} I_2 R_b, \tag{4.14}$$

where q is the electronic charge, R_i is the photodetector responsivity, I_2 is a noise-bandwidth factor, and R_b is the system bit rate (Kahn; Barry, 1997).

The thermal noise variance for a FET-based transimpedance preamplifier contains white noise from the feedback resistor, FET white channel noise, and FET $1/f$ channel noise. It is expressed as (Kahn; Barry, 1997):

$$\sigma^2_{\mathrm{thermal},i} = \frac{4k_{\mathrm{B}}T}{R_F} I_2 R_b + \frac{16\pi^2 k_{\mathrm{B}} T}{g_m} \left(\Gamma + \frac{1}{g_m R_D} \right) C_T^2 I_3 R_b^3 + \frac{4\pi^2 K I_f C_T^2}{g_m} R_b^2, \tag{4.15}$$

where k_{B} is Boltzmann's constant, T is the absolute temperature, R_F is the feedback resistance, g_m is the FET transconductance, Γ is the FET channel noise factor, R_D is the drain resistance, C_T is the total input capacitance (often dominated by the large area of the photodetector), K is the FET $1/f$ (flicker) noise coefficient, and I_3 and I_f are additional noise-bandwidth factors.

The corresponding branch-voltage variance is obtained by scaling the total input-referred current noise variance with the TIA gain:

$$\sigma_{v,i}^2 = G_{\mathrm{T}}^2 \, \sigma_{\mathrm{total},i}^2. \tag{4.16}$$

This formulation separates the two modeling layers cleanly: the reference parameters contribute physically to the current-noise terms, whereas G_{T} acts as the gain that converts the total input-referred current noise into a voltage-domain noise level.

At the branch level, the receiver-noise vector at the Analog-to-Digital Converter (ADC) input can be written as:

$$\mathbf{n} = [n_{\mathsf{R}} \; n_{\mathsf{G}} \; n_{\mathsf{B}}]^{\mathsf{T}}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_v), \tag{4.17}$$

with $\mathbf{\Sigma}_v = \mathrm{diag}(\sigma_{v,\mathsf{R}}^2, \sigma_{v,\mathsf{G}}^2, \sigma_{v,\mathsf{B}}^2)$. This branch-dependent model is physically meaningful, but it is more detailed than required for the detector comparison carried out later.

For the numerical experiments, the Monte Carlo pipeline therefore uses an effective isotropic Gaussian voltage noise model:

$$\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{n}}^2 \mathbf{I}_3), \tag{4.18}$$

where σ_{n} is the single simulation-noise parameter swept in the experiments. In this chapter, σ_{n} should be interpreted as a calibrated voltage-domain surrogate for the total front-end noise, rather than as a separate physical source:

$$\mathbf{n}_{\mathsf{CE}} = \hat{\mathbf{H}}^{-1} \mathbf{n}. \tag{4.19}$$

With these equations, both the physical and the simulation-level descriptions of the receiver are complete. The next section builds decision rules on top of this model, starting from the conventional minimum-distance baseline and then moving to data-driven detectors that can adapt to the warped practical constellation.

4.3 Learning-Based Detection Framework

As discussed in Section 4.2.5, the MED relies on fixed Voronoi regions determined by the nominal constellation. Section 4.2 shows, however, that practical OTS components warp the symbol geometry and that linear equalization does not fully

restore the ideal IEEE decision regions. Under these conditions, the fixed decision boundaries of MED become increasingly suboptimal as constellation order grows.

This chapter therefore formulates symbol detection as a supervised multiclass classification task. Let $\mathbf{V} \in \mathbb{R}^3$ denote the received voltage vector and let $m \in \{1, \ldots, M\}$ denote the index of the transmitted CSK symbol, where M is the constellation order. Given a labeled training set, a machine-learning model is trained to approximate the decision function:

$$f : \mathbb{R}^3 \to \{1, \ldots, M\}, \tag{4.20}$$

so that the predicted index $\hat{m} = f(\mathbf{V})$ reproduces the transmitted symbol from distorted receiver observations. Rather than imposing the ideal decision regions a priori, the classifier learns them directly from data generated by the practical transmitter–channel–receiver chain.

The resulting decision regions are warped but still structured: they are difficult to derive analytically in closed form, yet they remain learnable from representative data. Three data-driven alternatives to MED are therefore considered: kNN, FNNs, and KANs.

4.3.1 K-Nearest Neighbors Classifier

The kNN classifier is adopted as the first learned baseline because it imposes almost no parametric structure on the decision regions. It assigns a symbol index to each test sample by majority vote among its k nearest neighbors in the training set. Formally, let $\mathcal{N}_k(\mathbf{V})$ denote the set of k training samples closest to the test observation $\mathbf{V}$ under the Euclidean metric. The predicted symbol is (Goodfellow; Bengio; Courville, 2016):

$$\hat{m} = \arg \max_{j \in \{1,\ldots,M\}} \sum_{\mathbf{V}_i \in \mathcal{N}_k(\mathbf{V})} w_i \, I(m_i = j), \tag{4.21}$$

where $I(\cdot)$ is the indicator function and w_i is a weighting factor. The indicator function evaluates to 1 if the condition inside the parentheses is true (i.e., if the i-th neighbor belongs to the candidate class j) and 0 otherwise, effectively filtering and tallying the votes. Under uniform weighting, $w_i = 1$ for all neighbors, meaning the summation simply counts the raw number of neighbors belonging to class j; under distance weighting, $w_i = 1/\|\mathbf{V} - \mathbf{V}_i\|_2$, which gives closer neighbors a stronger influence on the decision.

Unlike MED, kNN adapts directly to the empirical geometry of the received clusters. Its main drawback is inference cost, which scales with the size of the stored training set.

4.3.2 Feedforward Neural Network

A FNN is adopted as the main neural baseline. It constructs the classifier through successive affine transformations followed by fixed nonlinear activation functions. For a layer l, the hidden representation is given by (Goodfellow; Bengio; Courville, 2016):

$$\mathbf{a}^{(l+1)} = g\Big(\mathbf{W}^{(l)}\mathbf{a}^{(l)} + \mathbf{b}^{(l)}\Big), \tag{4.22}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are learnable parameters and $g(\cdot)$ is a fixed activation. In this study, the FNN is included because it can represent nonlinear class boundaries without imposing any explicit geometric model on the received constellation.

4.3.3 Kolmogorov–Arnold Network

In contrast to FNNs, a KAN replaces fixed node activations by learnable univariate functions associated with the connections between neurons. Its layer-to-layer transformation can be written as (Liu *et al.*, 2025; Qiu *et al.*, 2025):

$$x_i^{(l+1)} = \sum_{j=1}^{n_l} \phi_{l,j,i}\Big(x_j^{(l)}\Big), \tag{4.23}$$

where each $\phi_{l,j,i}(\cdot)$ is a trainable nonlinear function. Because the nonlinearity resides in the edges rather than in the nodes, a KAN can potentially match the inference quality of a conventional FNN with fewer total parameters. This property is particularly appealing for practical CSK receivers, where computational and memory budgets are constrained. Moreover, the distortion introduced by practical optical devices is smooth but nonlinear, so a model whose functional basis can adapt to the data may capture the warped decision regions more efficiently than a network with fixed activations.

The specific KAN variant adopted in this work follows the ReLU-KAN formulation of Qiu *et al.* (2025). Each layer replaces the standard B-spline basis functions by pairwise ReLU products, yielding $g + k$ basis activations per input dimension, where g is the grid resolution and k is the basis order. These activations are arranged as a 2-D feature map and processed by a convolution kernel of size $(g + k) \times n_{\text{in}}$, where n_{in} is the number of input neurons. The output of each layer is therefore

$$\mathbf{h}^{(l+1)} = \mathrm{Conv}\Big(\Big[\mathrm{ReLU}(\mathbf{h}^{(l)} - \boldsymbol{a}) \cdot \mathrm{ReLU}(\boldsymbol{b} - \mathbf{h}^{(l)})\, r\Big]^2\Big), \tag{4.24}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are phase-boundary parameters and $r = 4g^2/(k+1)^2$ is a normalization constant. This formulation is attractive in the present context because it preserves compatibility with efficient GPU operations while allowing the nonlinearity itself to adapt to the structure of the data.

4.3.4 Comparison Methodology

The objective is not to replace model-based detection indiscriminately, but to determine under which conditions learned detectors provide a tangible advantage over a conventional geometric rule. Accordingly, MED quantifies the performance attainable with the nominal constellation geometry, whereas kNN, FNN, and KAN quantify how much of the practical mismatch can be recovered when the decision rule is learned from simulated observations. The simulation framework and the resulting hyperparameters are reported together with the numerical evidence in the next section, instead of being separated from it.

4.4 Simulation Framework and Results

This section establishes the experimental framework based on the theoretical model developed in Section 4.2, followed by a comparative analysis of the detector results. The complete numerical pipeline was built in Python: NumPy and SciPy handled the physical-layer computations, `colour-science` managed the CIE 1931 data, scikit-learn was used for classical machine learning baselines, and PyTorch powered the neural network models.

4.4.1 Dataset Generation and Noise Calibration

The dataset-generation procedure follows directly from Section 4.2. For each symbol, the practical RGB power vector is obtained from the OTS constellation mapping, converted to drive currents through Eq. (4.3), propagated through the channel matrix $\mathbf{H}$, and finally perturbed by additive voltage noise according to Eq. (4.18). Each sample is therefore a labeled voltage triplet (V_1, V_2, V_3).

For 4-CSK and 8-CSK, the normalized RGB coefficients are taken from Table 6. For 16-CSK, the coefficients are computed once by solving

$$\mathbf{P}_k^* = \arg \min_{\mathbf{P} \in [0,1]^3} \left\| \mathbf{s}_{\mathrm{ref},k} - \mathbf{s}_{\mathbf{P}} \right\|_2 , \tag{4.25}$$

using Powell optimization with an equal-power initialization and 2 000 maximum iterations. The optimized vectors are then reused across all Monte Carlo runs.

Tables 7, 8, and 9 consolidate the settings that matter for interpreting the results.

Section 4.2.6 defines the physical noise terms in current form and the simulation noise in voltage form. The present subsection evaluates those expressions numerically using the actual spectral data of the optical filters and photodetector from the simulation model, thereby establishing per-channel physical references for the noise sweep.

Unlike a single spectrally flat approximation, the background photocurrent depends on the spectral overlap between each optical filter and the photodetector re-

Table 7 – System-setup parameters.

Parameter	**Value**
CSK orders	4, 8, and 16
Random seed	42
Spectral grid	380–780 nm, 1 nm spacing
Link distance	1 m
LED spacing	25 mm
Lambertian mode	$m = 10$
Photodiode	PD204-6C, $A_{\mathrm{pd}} = 7.069$ mm^2
TIA gain	$G_{\mathrm{T}} = 10^4$ V/A

Table 8 – Noise-model and calibration parameters (Kahn; Barry, 1997).

Parameter	**Value**
Ambient-light model	
Spectral irradiance	$p_n = 6$ mW/(cm^2·nm)
Background photocurrent	$I_{\mathrm{bg},i} = A_{\mathrm{pd}}\, p_n \int T_i(\lambda)\, R(\lambda)\, d\lambda$
TIA / FET front-end	
Feedback resistance	$R_F = 10$ kΩ
Transconductance	$g_m = 40$ mS
Gate capacitance	$C_T = 1.0$ pF
Drain current	$I_D = 20$ mA
$1/f$ noise coefficient	$K = 294$ fA
Channel-noise factor	$\Gamma = 2/3$
Temperature	$T = 300$ K
Calibration results at $B_{\mathrm{ref}} = 10$ MHz	
Red channel	$I_{\mathrm{bg}} = 58.79$ mA, $\sigma_v \approx 4.32$ mV
Green channel	$I_{\mathrm{bg}} = 25.72$ mA, $\sigma_v \approx 2.86$ mV
Blue channel	$I_{\mathrm{bg}} = 25.06$ mA, $\sigma_v \approx 2.82$ mV
Simulation noise	
Abstraction	$\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_3)$
Sweep range	$\sigma_{\mathrm{n}} \in [0, 10]$ mV
Operating points	$\sigma_{\mathrm{n}} = 5.9$ mV

sponsivity. For channel $i \in \{\mathsf{R}, \mathsf{G}, \mathsf{B}\}$, the ambient-light-induced photocurrent is:

$$I_{\mathrm{bg},i} = A_{\mathrm{pd}}\, p_n \int_{380}^{780} T_i(\lambda)\, R(\lambda)\, d\lambda, \tag{4.26}$$

where $p_n = 6$ mW/(cm^2·nm) is the isotropic bright-skylight spectral irradiance (Kahn; Barry, 1997), $T_i(\lambda)$ is the optical-filter transmittance, $R(\lambda)$ is the PD204-6C responsivity, and $A_{\mathrm{pd}} = 7.069$ mm^2 is its active area. The integral is evaluated numerically on the 1 nm grid used throughout the simulation. Because the red filter has the broadest passband and coincides with the peak of $R(\lambda)$, it collects the most ambient light: $I_{\mathrm{bg},\mathsf{R}} = 58.79$ mA, $I_{\mathrm{bg},\mathsf{G}} = 25.72$ mA, and $I_{\mathrm{bg},\mathsf{B}} = 25.06$ mA.

Table 9 – Dataset-generation protocol.

Parameter	**Value**
Main dataset sizes	10 000 train / 2 000 validation/ 2 000 test
Cross-validation	5 folds, 300 samples per symbol
Training-size sweep	10–3 000 samples per symbol

Evaluating Eq. (4.14) with $R_i P_{n,i} = I_{\mathrm{bg},i}$ and the effective noise bandwidth $B_{\mathrm{ref}} = 10$ MHz, the per-channel shot-noise variance is

$$\sigma^2_{\mathrm{shot},i} = 2q\, I_{\mathrm{bg},i}\, B_{\mathrm{ref}}. \tag{4.27}$$

After scaling by G_{T}, this yields voltage-domain shot-noise levels of $\sigma_{\mathrm{shot,R}} \approx 4.32$ mV, $\sigma_{\mathrm{shot,G}} \approx 2.86$ mV, and $\sigma_{\mathrm{shot,B}} \approx 2.82$ mV.

The thermal noise, which is independent of the optical channel, comprises four input-referred contributions at B_{ref}: feedback-resistor Johnson noise (≈ 0.041 mV), FET channel noise (< 0.001 mV), gate-leakage drain shot noise (≈ 0.002 mV), and $1/f$ noise (< 0.001 mV), all referred to the TIA output through G_{T}. The aggregate thermal term is $\sigma_{\mathrm{thermal}} \approx 0.041$ mV; more than two orders of magnitude below the shot noise in every channel.

The total per-channel voltage noise is therefore dominated by the shot-noise term. From Eqs. (4.13) and (4.16),

$$\sigma_{v,i} = G_{\mathrm{T}} \sqrt{\sigma^2_{\mathrm{shot},i} + \sigma^2_{\mathrm{thermal}}}, \tag{4.28}$$

giving $\sigma_{v,\mathrm{R}} \approx 4.32$ mV, $\sigma_{v,\mathrm{G}} \approx 2.86$ mV, and $\sigma_{v,\mathrm{B}} \approx 2.82$ mV. The worst-case channel (red) sets the physical noise floor at approximately 4.3 mV for the adopted front-end parameters.

Taking the root sum of squares across the three branches gives the combined noise floor $\sigma_{\mathrm{comb}} = \sqrt{\sigma^2_{v,\mathrm{R}} + \sigma^2_{v,\mathrm{G}} + \sigma^2_{v,\mathrm{B}}} \approx 5.90$ mV. Fig. 22 summarises the shot, thermal, and total noise contributions per channel together with the combined value at B_{ref}.

The adopted simulation sweep $\sigma_{\mathrm{n}} \in [0, 10]$ mV therefore spans from near-pristine conditions through the per-channel physical noise floor (≈ 2.8–4.3 mV) and well beyond the combined floor (≈ 5.9 mV). The chosen operating point ($\sigma_{\mathrm{n}} = 5.9$ mV) coincides with the root-sum-of-squares combined noise floor, thereby providing the most physically grounded single-parameter approximation of the full three-channel front-end noise for all experiments that follow.

The machine-learning training setup is summarized separately in Table 10 because it is easier to read when decoupled from the physical-layer protocol.

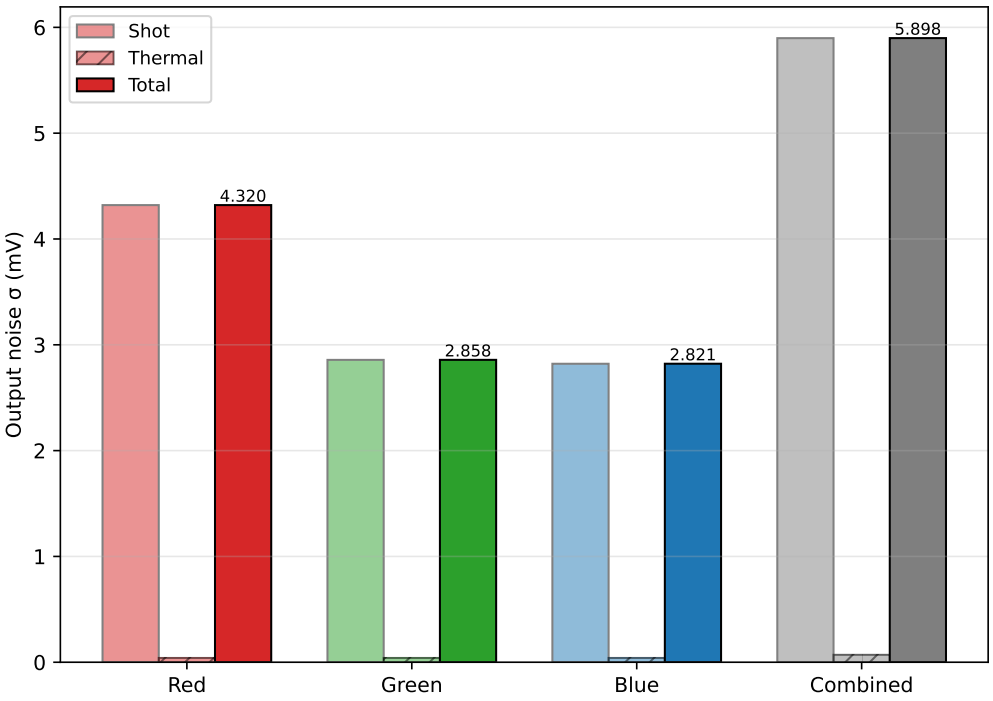


Figure 22 – Per-channel and combined noise at the reference operating point $B_{\mathrm{ref}} = 10$ MHz. Shot noise (hatched) dominates in every channel; the thermal contribution is negligible. The grey bars show the root-sum-of-squares combination across all three branches.

4.4.2 Hyperparameter Selection

Table 11 lists the best hyperparameter configurations selected by the 5-fold cross-validation procedure for each constellation order, using 1 000 samples per symbol at the operating-point noise $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

Several trends emerge from these results. Model complexity generally scales with constellation order, though not uniformly across classifier families. For kNN, $k = 5$ suffices for the well-separated 4-CSK constellation, whereas the denser 8-CSK and 16-CSK cases both select $k = 21$, consistent with the greater overlap among decision regions as the constellation becomes denser. The FNN architecture follows a non-monotonic trajectory: a two-layer network of 192 neurons with learning rate 10^{-4} and dropout 0.2 is selected for 4-CSK, reflecting the benefit of regularization even for the simplest problem when the search space is broad; a lighter two-layer topology of 96 neurons with no dropout prevails for 8-CSK; and a single hidden layer of 128 neurons with dropout 0.2 is chosen for 16-CSK, suggesting that wider single-layer representations generalize better when the number of classes is large. The ReLU-KAN converges on compact single-layer topologies across all orders, $[32]$ for both 4-CSK and 16-CSK, and $[128]$ for 8-CSK, all at grid resolution $g = 3$ and spline order $k = 3$. The preference for single hidden layers is notable: the learnable per-edge activation

Table 10 – Machine-learning training configuration used in the CSK study.

Training item	**Configuration**
Input features	Received voltage triplet (V_1, V_2, V_3)
Target labels	CSK symbol classes for $M \in \{4, 8, 16\}$
Preprocessing	Feature standardization to zero mean and unit variance before neural-network training
Neural optimizers	Adam for both FNN and ReLU-KAN
Default learning rate	10^{-3} for both neural models
Batch size	128
Main training budget	100 epochs for FNN and ReLU-KAN
Cross-validation budget	50 epochs during hyperparameter search to reduce computational cost
Loss function	Multiclass cross-entropy
Candidate FNN architectures	$[32]$, $[64]$, $[128]$, $[64, 32]$, $[128, 64]$, $[256, 128]$, $[64, 32, 16]$, $[128, 64, 32]$, and $[256, 128, 64]$ with learning rate in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and dropout in $\{0, 0.1, 0.2, 0.3\}$
Candidate ReLU-KAN architectures	Same nine topologies as FNN with grid $g \in \{3, 5, 8\}$ and order $k \in \{2, 3\}$
Selected kNN search space	$k \in \{1, 3, 5, 7, 9, 11, 15, 21\}$ with uniform or distance weighting
Model selection criterion	Mean 5-fold validation accuracy for each CSK order

Table 11 – Best hyperparameters selected via 5-fold cross-validation for each CSK order.

Classifier	**Parameter**	**4-CSK**	**8-CSK**	**16-CSK**
kNN	Neighbors k	5	21	21
	Weighting	Uniform	Uniform	Uniform
	Cross-validation accuracy	0.9975	0.8645	0.7299
FNN	Hidden layers	[128, 64]	[64, 32]	[128]
	Learning rate	10^{-4}	10^{-3}	10^{-3}
	Dropout	0.2	0.0	0.2
	Cross-validation accuracy	0.9975	0.8700	0.7392
ReLU-KAN	Hidden layers	[32]	[128]	[32]
	Grid g	3	3	3
	Spline order k	3	3	3
	Cross-validation accuracy	0.9938	0.8676	0.7361

functions provide sufficient representational capacity without the additional depth that FNNs sometimes require.

Across all classifiers, the cross-validation accuracy drops monotonically with increasing M, from roughly 0.99 for 4-CSK down to about 0.73 for 16-CSK. This decline underscores the progressively warped and densely packed constellation geometry, which imposes escalating demands on the decision boundary even with a modest training set.

Table 12 compares the total number of trainable parameters of the two neural architectures for each constellation order. The FNN counts include the weights and biases of all linear layers plus the batch-normalization scale and shift parameters; the ReLU-KAN counts include the learnable phase boundaries, the convolutional kernel, and the linear classification head. For 4-CSK the two-layer FNN ([128, 64]) contains 9 412 parameters, the largest model in the comparison, whereas the single-layer ReLU-KAN ([32]) requires only 776, roughly 12 times fewer. For 8-CSK the counts are closer: 2 792 (FNN) versus 3 500 (ReLU-KAN), because the KAN selects a wider hidden layer ([128]) while the FNN opts for [64, 32]. For 16-CSK the ReLU-KAN is again more compact at 1 172 versus 2 832 for the FNN. In all cases both architectures are extremely lightweight, on the order of a few thousand parameters, which is consistent with the low intrinsic dimensionality of the three-voltage input space.

By contrast, kNN stores the entire training set as its model. At the evaluation training budget of 10 000 samples this amounts to 30 000 stored floating-point values, an order of magnitude larger than either neural network, although no gradient-based optimization is involved. The MED has zero trainable parameters; its decision rule is derived entirely from the nominal constellation coordinates and the channel model.

Table 12 – Trainable parameter count for the neural-network-based detectors at each constellation order.

Classifier	4-CSK	8-CSK	16-CSK
FNN	9 412	2 792	2 832
ReLU-KAN	776	3 500	1 172

4.4.3 Classification Accuracy versus Noise Level

Table 13 reports the classification accuracy of all four detectors across five representative noise levels and the operating-point noise for three constellation orders. These values come from the dedicated noise-sweep experiments, which use 1 000 training and 1 000 test samples per symbol together with the cross-validation-selected hyperparameters.

Fig. 23 shows the complete accuracy-versus-noise curves, covering all noise levels from $\sigma_{\mathrm{n}} = 1$ to 10 mV. Several points deserve emphasis in these results:

MED performance and the impact of constellation distortion.

The MED detector suffers a steep accuracy loss as the constellation order grows. For 4-CSK it remains near-ideal up to several millivolts of noise, still reaching 97.8% at $\sigma_{\mathrm{n}} = 5$ mV. At the same noise level, however, 8-CSK accuracy drops to 80.4% and 16-CSK collapses to just 48.5%. The severity of the constellation distortion

Table 13 – Classification accuracy for each detector across noise levels and constellation orders. Best result per row in **bold**.

M	σ_{n} (mV)	MED	KNN	FNN	ReLU-KAN
4	1	1.0000	**1.0000**	**1.0000**	**1.0000**
	3	0.9990	**1.0000**	**1.0000**	**1.0000**
	5	0.9775	**1.0000**	0.9995	0.9995
	5.9	0.9515	**0.9970**	**0.9970**	**0.9970**
	7	0.9175	**0.9920**	**0.9920**	0.9910
	10	0.8165	0.9390	**0.9480**	0.9420
8	1	0.9990	**1.0000**	**1.0000**	**1.0000**
	3	0.9250	**0.9815**	**0.9815**	0.9810
	5	0.8040	0.9040	0.9045	**0.9065**
	5.9	0.7515	0.8640	**0.8645**	0.8640
	7	0.6930	0.8075	**0.8105**	0.8095
	10	0.5595	0.6475	0.6540	**0.6615**
16	1	0.7090	**0.9830**	0.9815	**0.9830**
	3	0.5935	0.8995	0.8935	**0.9025**
	5	0.4850	0.7760	0.7710	**0.7790**
	5.9	0.4395	0.7195	**0.7265**	0.7235
	7	0.3975	0.6385	**0.6455**	0.6370
	10	0.3160	0.4745	**0.4935**	0.4865

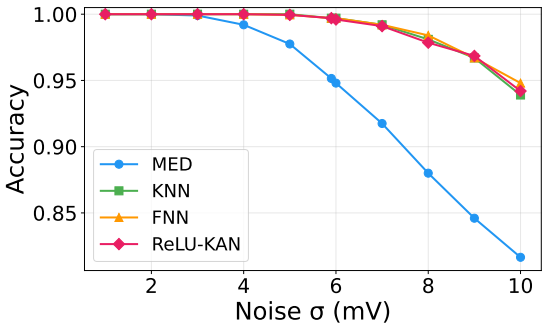
is best illustrated by the 16-CSK case ($\sigma_{\mathrm{n}} = 1$ mV): MED achieves only 70.9%, meaning that nearly 30% of the symbols are misclassified even under negligible noise – a direct consequence of the practical OTS constellation geometry deviating from the nominal IEEE reference positions.

Learned classifiers versus MED.

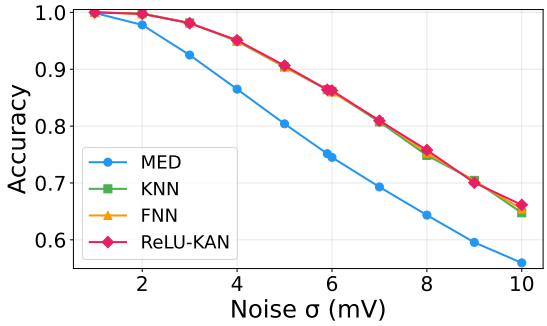
All three learned classifiers consistently surpass MED at every noise level and constellation order examined. At $\sigma_{\mathrm{n}} = 10$ mV, the best learned detector cuts the symbol-error rate relative to MED by roughly 72% for 4-CSK, 23% for 8-CSK, and 26% for 16-CSK. Just as critically, every learned classifier achieves accuracies above 98% at $\sigma_{\mathrm{n}} = 1$ mV for 16-CSK, where MED stagnates at 70.9%, confirming that data-driven decision boundaries are indispensable once the practical constellation geometry departs substantially from the nominal design.

Relative ranking among learned classifiers.

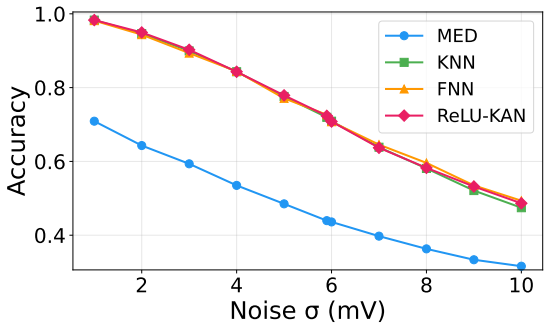
No single learned model dominates uniformly; the best-performing classifier varies with both the constellation order and the noise regime. For 4-CSK, kNN scores the highest or tied-highest accuracy throughout most of the noise range. At $\sigma_{\mathrm{n}} = 5$ mV, kNN achieves a perfect 100% while the neural models reach 99.95%. At the harshest noise ($\sigma_{\mathrm{n}} = 10$ mV), however, the FNN takes the lead with 94.8% versus 93.9% for kNN,



(a) 4-CSK



(b) 8-CSK



(c) 16-CSK

Figure 23 – Classification accuracy versus noise standard deviation σ_{n} for (a) 4-CSK, (b) 8-CSK, and (c) 16-CSK. Each curve represents one of the four detectors evaluated: MED, KNN, FNN, and ReLU-KAN.

supporting the observation that smooth parametric boundaries eventually outperform purely instance-based rules in high-noise regimes.

For 8-CSK, ReLU-KAN emerges as a competitive alternative: it achieves the best accuracy at $\sigma_{\mathrm{n}} = 5\,\mathrm{mV}$ (90.65% versus 90.45% for FNN and 90.40% for kNN), suggesting that its learnable per-edge nonlinearities capture the eight-class geometry faithfully when the noise begins to broaden the clusters. At the extreme noise of 10 mV, ReLU-KAN also leads with 66.15%, ahead of FNN (65.4%) and kNN (64.75%).

For 16-CSK, the picture is more nuanced. ReLU-KAN leads at low-to-moderate noise, achieving the top accuracy at $\sigma_{\mathrm{n}} \in \{1, 3, 5\}\,\mathrm{mV}$, whereas the FNN overtakes it from $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$ onward and ultimately reaches 49.35% at $\sigma_{\mathrm{n}} = 10\,\mathrm{mV}$, the highest among all models. Across all three constellations the inter-classifier gaps remain modest, typically within one to two percentage points, so the overarching conclusion is clear: any of the three data-driven detectors substantially outperforms geometric detection, while the choice of the best specific model is a second-order decision that depends on the operating regime.

4.4.4 Error Patterns Beyond Accuracy

Aggregate accuracy, while useful for ranking detectors, compresses all per-class error information into a single scalar and can therefore mask severe failure modes on individual symbols. A richer picture emerges from a suite of complementary metrics, each of which highlights a different aspect of classifier behavior:

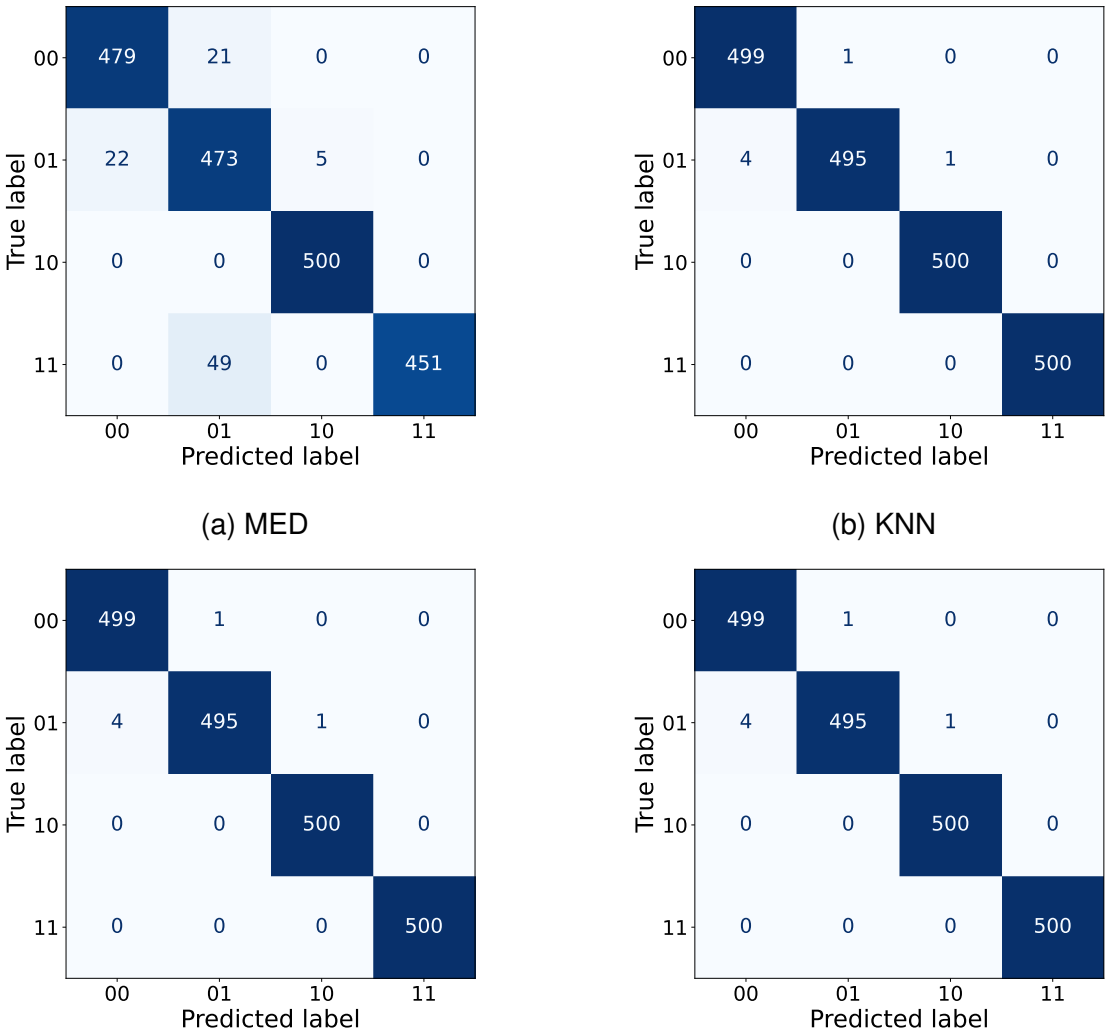
- A **confusion matrix** $\mathbf{C} \in \mathbb{N}^{M \times M}$ records the joint distribution of true and predicted labels: the entry C_{ij} counts how many samples of class i are classified as class j. A perfect detector produces a strictly diagonal matrix; off-diagonal mass reveals which symbol pairs are most frequently confused.
- **Precision** for class i is the fraction of samples predicted as i that truly belong to i, i.e. $P_i = C_{ii} / \sum_j C_{ji}$, while **recall** is the fraction of true class-i samples that are correctly recovered, $R_i = C_{ii} / \sum_j C_{ij}$. The **F1-score** $F_{1,i} = 2P_iR_i/(P_i + R_i)$ is their harmonic mean and penalizes any imbalance between the two. Macro-averaging over all M classes treats every symbol equally, making the metric sensitive to failure on rare or hard classes.
- A **Receiver Operating Characteristic (ROC)** curve plots the true-positive rate against the false-positive rate as a decision threshold is swept (Fawcett, 2006). For a multiclass problem each class is evaluated in a one-versus-rest fashion, and the **Area Under Curve (AUC)** summarizes the result as a single scalar: 1.0 indicates perfect class separability regardless of threshold, 0.5 corresponds to random guessing.

Confusion matrices.

Figs. 24–26 present the confusion matrices for each detector at the operating-point noise.

For 4-CSK (Fig. 24), all three learned classifiers yield nearly diagonal matrices; the MED likewise performs well, with its only visible off-diagonal mass concentrated on the symbol pair `01`↔`11`, which share the closest chromaticity coordinates in the distorted constellation. Reading the MED matrix row by row, symbol `01` loses roughly 5% of its samples to `11`, while `11` sheds about 10% back to `01`, a reciprocal confusion driven by the proximity of these two clusters in the received voltage space. The learned detectors eliminate this residual confusion almost entirely.

As the constellation density increases to 8-CSK (Fig. 25), the MED matrix develops broad off-diagonal bands. The most prominent confusion arises for symbol `111`, whose column absorbs samples from `000`, `001`, and `011`; in the opposite direction, `001`



(a) MED

(b) KNN

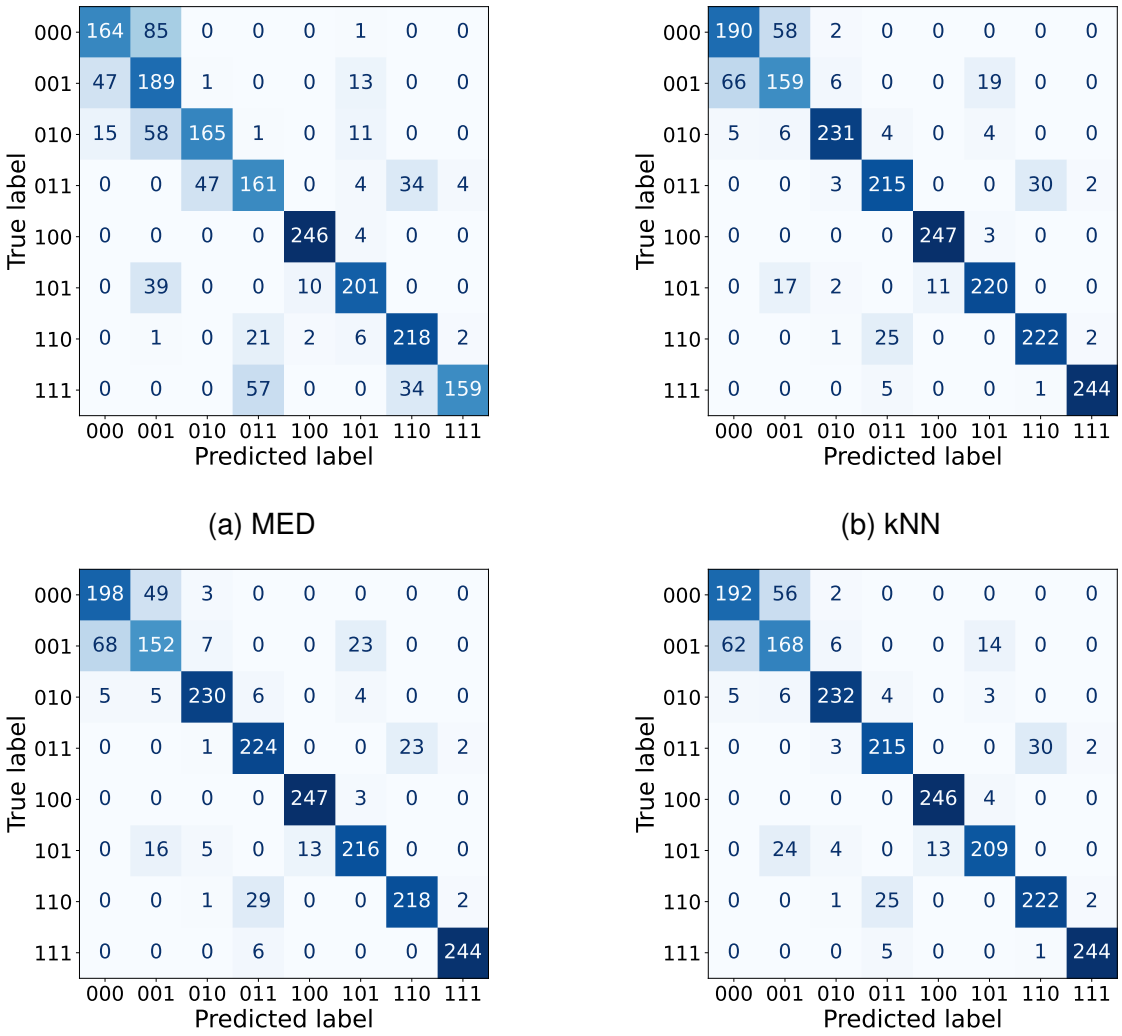
(c) FNN

(d) ReLU-KAN

Figure 24 – Confusion matrices for 4-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

receives substantial spillover from `000` and `110`. These patterns indicate that the constellation distortion pulls several cluster centroids into overlapping regions, causing the fixed Voronoi boundaries of MED to bisect them. The learned detectors retain markedly sharper diagonal structure: the worst per-class recall for any learned classifier is approximately 64% (kNN on `001`), compared with 64% for MED on the same class—but the crucial difference is that the learned models maintain high recall on the remaining classes, whereas the MED degrades broadly.

The disparity is most acute for 16-CSK (Fig. 26). The MED exhibits widespread class aliasing: symbols `1100` and `1110` are almost never detected correctly (recall of only 2.4% and 15.2%, respectively), with their samples scattered across multiple unrelated classes. Symbol `1111` likewise suffers a recall of just 36.8%, with 274 of its 750 true samples absorbed by neighboring classes. By contrast, the FNN and kNN confine the residual errors to a handful of geometrically adjacent symbol pairs: for example,



(a) MED

(b) kNN

(c) FNN

(d) ReLU-KAN

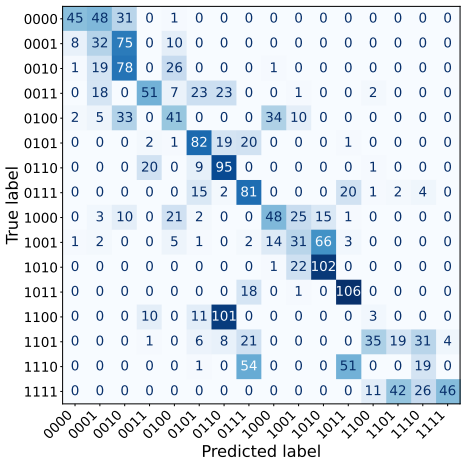
Figure 25 – Confusion matrices for 8-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

the FNN's worst recall is 21.6% (on `0001`), but most classes exceed 65%, producing a far more balanced error profile. The learned models thus improve not only aggregate accuracy but also the *balance* of errors across the symbol alphabet, a property that matters for communication systems, where heavily penalized symbols can dominate the overall bit-error rate.

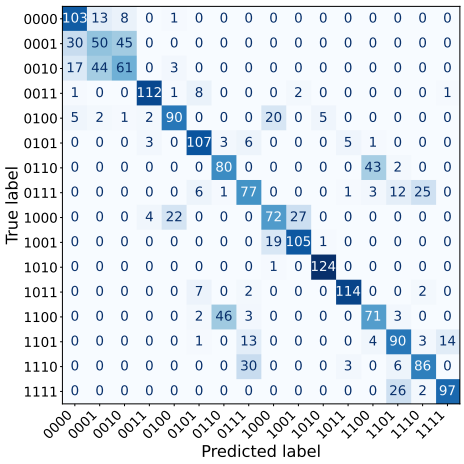
Per-class precision, recall, and F1 scores.

Tables 14–16 report the per-symbol precision (P), recall (R), and F1-score achieved by each detector for the 4-, 8-, and 16-CSK constellations at the operating-point noise level $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$. These tables provide a detailed view of the classification performance for each individual symbol.

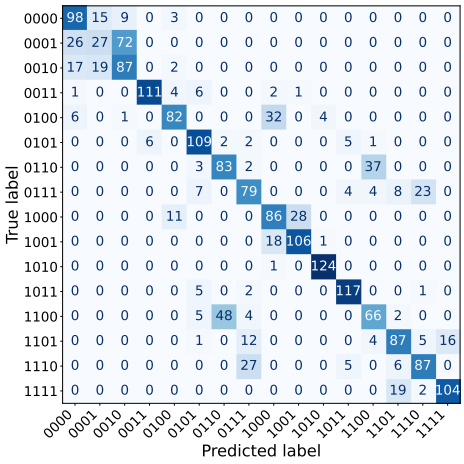
To complement this symbol-level analysis, Table 17 reports the macro-averaged precision, recall, and F1-score for each classifier and modulation order, where each



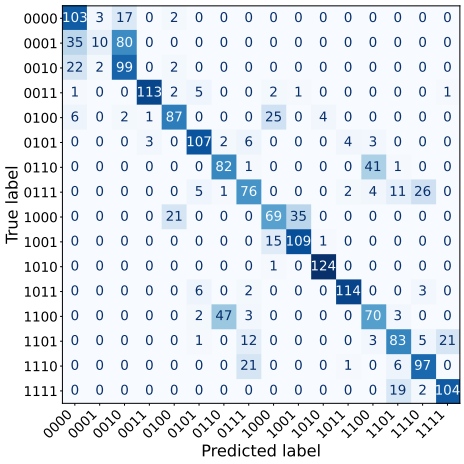
(a) MED



(b) KNN



(c) FNN



(d) ReLU-KAN

Figure 26 – Confusion matrices for 16-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

quantity is obtained by taking the arithmetic mean across all symbol classes. This separation keeps the classwise precision–recall discussion distinct from the threshold-swept ROC analysis presented later.

Table 14 – Per-symbol precision (P), recall (R), and F1-score for 4-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

	MED			**KNN**			**FNN**			**ReLU-KAN**		
Symbol	P	R	F1	P	R	F1	P	R	F1	P	R	F1
00	0.956	0.958	0.957	0.992	0.998	0.995	0.992	0.998	0.995	0.992	0.998	0.995
01	0.871	0.946	0.907	0.998	0.990	0.994	0.998	0.990	0.994	0.998	0.990	0.994
10	0.990	1.000	0.995	0.998	1.000	0.999	0.998	1.000	0.999	0.998	1.000	0.999
11	1.000	0.902	0.948	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

The F1 metric is more revealing than raw accuracy when classification errors are unevenly distributed across symbols. While Tables 14–16 present the detailed per-symbol precision, recall, and F1 values, Table 17 condenses the same information into macro averages so that the reader can compare overall detector balance without mixing those summaries with the separate ROC-based threshold analysis.

Table 15 – Per-symbol precision (P), recall (R), and F1-score for 8-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

	MED			KNN			FNN			ReLU-KAN		
Symbol	P	R	F1	P	R	F1	P	R	F1	P	R	F1
000	0.726	0.656	0.689	0.728	0.760	0.744	0.731	0.792	0.760	0.741	0.768	0.754
001	0.508	0.756	0.608	0.663	0.636	0.649	0.685	0.608	0.644	0.661	0.672	0.667
010	0.775	0.660	0.713	0.943	0.924	0.933	0.931	0.920	0.926	0.935	0.928	0.932
011	0.671	0.644	0.657	0.863	0.860	0.862	0.845	0.896	0.870	0.863	0.860	0.862
100	0.953	0.984	0.969	0.957	0.988	0.972	0.950	0.988	0.969	0.950	0.984	0.967
101	0.838	0.804	0.820	0.894	0.880	0.887	0.878	0.864	0.871	0.909	0.836	0.871
110	0.762	0.872	0.813	0.877	0.888	0.883	0.905	0.872	0.888	0.877	0.888	0.883
111	0.964	0.636	0.766	0.984	0.976	0.980	0.984	0.976	0.980	0.984	0.976	0.980

Table 16 – Per-symbol precision (P), recall (R), and F1-score for 16-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

	MED			KNN			FNN			ReLU-KAN		
Symbol	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0000	0.789	0.360	0.495	0.660	0.824	0.733	0.662	0.784	0.718	0.617	0.824	0.705
0001	0.252	0.256	0.254	0.459	0.400	0.427	0.443	0.216	0.290	0.667	0.080	0.143
0010	0.344	0.624	0.443	0.530	0.488	0.508	0.515	0.696	0.592	0.500	0.792	0.613
0011	0.607	0.408	0.488	0.926	0.896	0.911	0.949	0.888	0.917	0.966	0.904	0.934
0100	0.366	0.328	0.346	0.769	0.720	0.744	0.804	0.656	0.722	0.763	0.696	0.728
0101	0.547	0.656	0.596	0.817	0.856	0.836	0.801	0.872	0.835	0.849	0.856	0.853
0110	0.383	0.760	0.509	0.615	0.640	0.627	0.624	0.664	0.643	0.621	0.656	0.638
0111	0.413	0.648	0.505	0.588	0.616	0.602	0.617	0.632	0.625	0.628	0.608	0.618
1000	0.490	0.384	0.430	0.643	0.576	0.608	0.619	0.688	0.652	0.616	0.552	0.582
1001	0.344	0.248	0.288	0.784	0.840	0.811	0.785	0.848	0.815	0.752	0.872	0.807
1010	0.557	0.816	0.662	0.954	0.992	0.973	0.961	0.992	0.976	0.961	0.992	0.976
1011	0.582	0.848	0.691	0.927	0.912	0.919	0.893	0.936	0.914	0.942	0.912	0.927
1100	0.057	0.024	0.034	0.582	0.568	0.575	0.589	0.528	0.557	0.579	0.560	0.569
1101	0.302	0.152	0.202	0.647	0.720	0.682	0.713	0.696	0.704	0.675	0.664	0.669
1110	0.238	0.152	0.185	0.729	0.688	0.708	0.737	0.696	0.716	0.729	0.776	0.752
1111	0.920	0.368	0.526	0.866	0.776	0.819	0.867	0.832	0.849	0.825	0.832	0.829

Table 17 – Macro-averaged precision, recall, and F1-score at the operating-point noise $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$. Best result per metric is in **bold**.

	4-CSK			8-CSK			16-CSK		
Classifier	P	R	F1	P	R	F1	P	R	F1
MED	0.9543	0.9515	0.9519	0.7746	0.7515	0.7544	0.4494	0.4395	0.4159
KNN	**0.9970**	**0.9970**	**0.9970**	0.8636	0.8640	0.8637	0.7185	0.7195	0.7176
FNN	**0.9970**	**0.9970**	**0.9970**	0.8636	**0.8645**	0.8634	0.7237	**0.7265**	**0.7204**
ReLU-KAN	**0.9970**	**0.9970**	**0.9970**	**0.8650**	0.8640	**0.8643**	**0.7306**	0.7235	0.7090

For the near-perfect 4-CSK case, the per-symbol results in Table 14 show that all learned classifiers achieve almost flawless detection. This translates into identical macro-averaged precision, recall, and F1 values of 0.997 for kNN, FNN, and ReLU-KAN in Table 17, clearly outperforming the MED, whose corresponding macro values remain near 0.95. Inspection of Table 14 shows that this MED degradation is mainly caused by the single difficult symbol pair (01 versus 00), whose confusion slightly lowers all three averages.

The gap widens markedly as the constellation density increases. According to the macro-averaged results in Table 17, the MED's performance drops to 0.449 in precision, 0.440 in recall, and 0.416 in F1 for the 16-CSK constellation, indicating that many symbols suffer from substantial detection errors. In contrast, the learned classi-

fiers maintain significantly higher performance. The FNN reaches the best macro recall and macro F1 at 0.7265 and 0.7204, respectively, while ReLU-KAN reaches the best macro precision at 0.7306. This confirms that the learned detectors improve the overall balance between missed detections and false alarms rather than merely boosting a single summary metric.

The detailed per-symbol results in Table 16 reveal the origin of this degradation. Several symbols become particularly difficult for the MED, notably $\texttt{1100}$, $\texttt{1110}$, and $\texttt{1101}$, which achieve F1 scores of only 0.034, 0.185, and 0.202, respectively. These classes correspond to chromaticity points that are pulled into a dense cluster by hardware distortion, making them difficult to separate using the distance-based MED rule. The learned detectors substantially mitigate this issue: for the same symbols, kNN achieves F1 scores of 0.575, 0.708, and 0.682, respectively.

The per-symbol tables also decompose the F1 score into its precision and recall components, revealing failure modes that aggregated metrics can conceal. In general, the learned classifiers improve both precision and recall relative to MED, rather than trading one for the other. A notable exception occurs for ReLU-KAN in the 16-CSK class $\texttt{0001}$, where the recall drops to only 0.080 despite a precision of 0.667, indicating that the network rarely assigns samples to this label. This near-collapse does not occur in either FNN (recall $= 0.216$) or kNN (recall $= 0.400$) for the same class, highlighting the value of evaluating multiple architectural families to identify potential class-specific blind spots.

ROC analysis.

A ROC curve plots the true-positive rate against the false-positive rate as the classification threshold is swept from its most conservative to its most permissive value (Fawcett, 2006). In this context, the true-positive rate is simply the recall of the target class, that is, the fraction of symbol-m samples correctly identified as symbol m. The false-positive rate is the fraction of all non-m samples that are incorrectly accepted as symbol m. Because CSK detection is a multiclass problem, each class is evaluated in a one-versus-rest fashion: for a given symbol m, all other symbols are grouped as the negative class, and the resulting binary ROC curve measures how well the detector separates symbol m from the remainder of the constellation. The AUC summarizes each curve with a single scalar; a value of 1.0 indicates perfect class separability regardless of threshold, whereas 0.5 corresponds to random guessing. To summarize an entire detector over all M classes, the chapter uses the macro-averaged one-versus-rest AUC, i.e., the arithmetic mean of the M classwise AUC values.

Figs. 27–29 show the one-versus-rest ROC curves for each constellation order, and Table 18 reports the corresponding macro-averaged AUC values. Stating this aver-

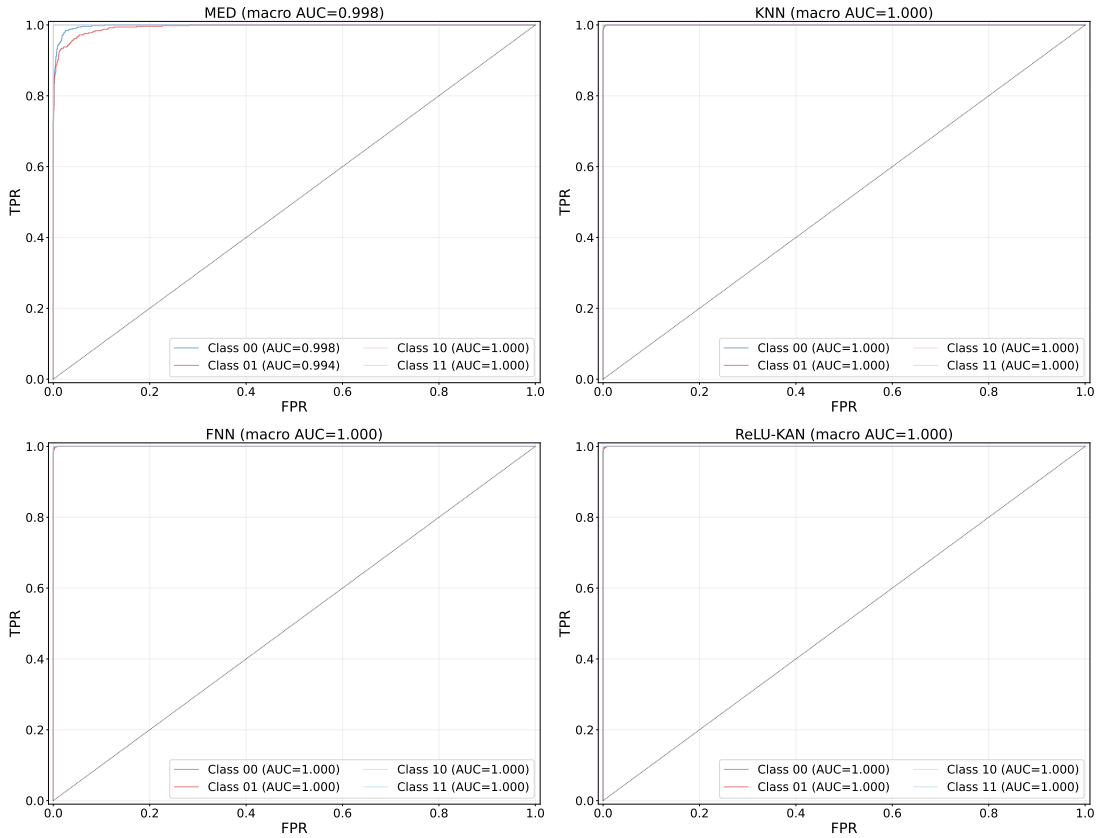


Figure 27 – One-versus-rest ROC curves for 4-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$. The vertical axis is the true-positive rate (recall), and the horizontal axis is the false-positive rate, i.e., the fraction of non-target symbols incorrectly accepted as the target class.

Table 18 – Macro-averaged one-versus-rest AUC at the operating-point noise $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$. Each entry is the arithmetic mean of the classwise AUC values in the corresponding ROC family. Best result per constellation order is in **bold**.

Classifier	**4-CSK**	**8-CSK**	**16-CSK**
MED	0.9980	0.9545	0.9249
KNN	**1.0000**	0.9856	0.9761
FNN	**1.0000**	**0.9890**	**0.9826**
ReLU-KAN	**1.0000**	0.9889	**0.9826**

age explicitly is important: a classwise AUC read from an individual curve can be lower than the macro AUC of the same detector because the table averages over all classes. This is why, for instance, a single difficult class in 8-CSK may yield an AUC below the detector's macro-average result.

The macro-averaged AUC values confirm the ranking established by accuracy yet reveal a subtlety: even for 16-CSK (Fig. 29), where the MED's classification accuracy is only about 44%, its macro AUC still reaches 0.9249, meaning that the detector retains substantial discriminative power that its fixed decision thresholds fail to exploit. The learned classifiers push the macro AUC above 0.97 for all constellation orders,

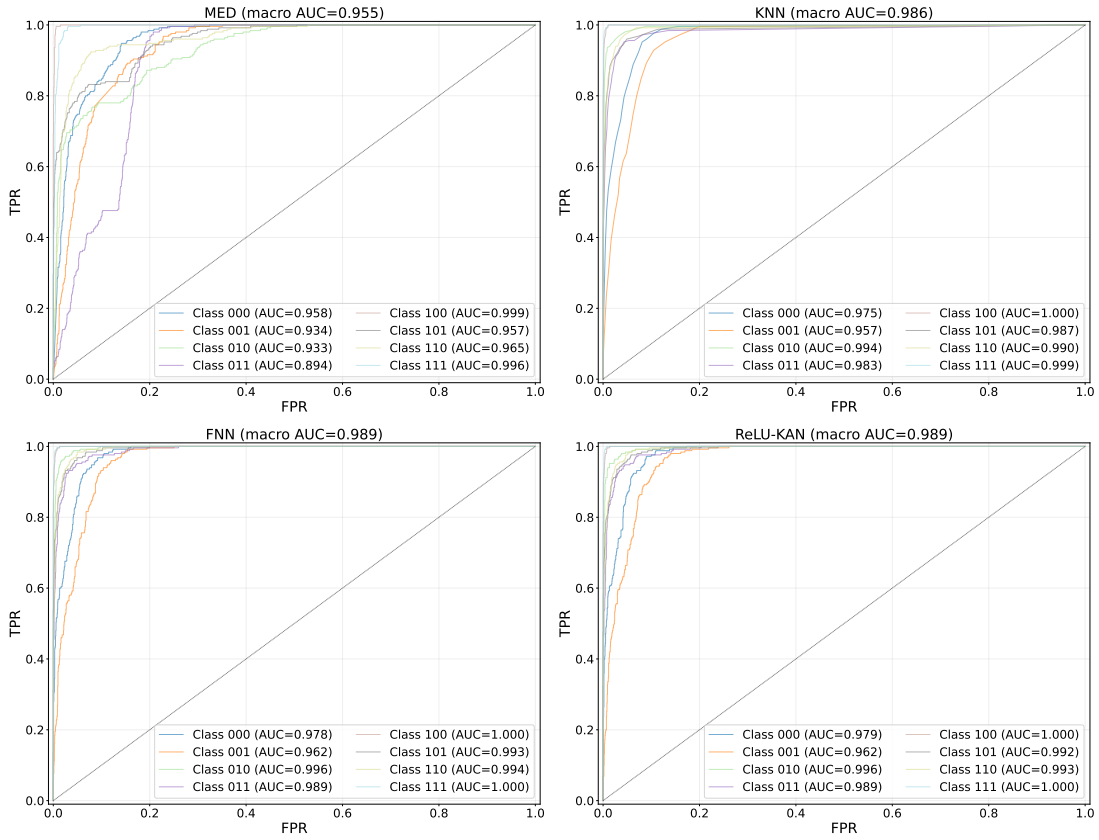


Figure 28 – One-versus-rest ROC curves for 8-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

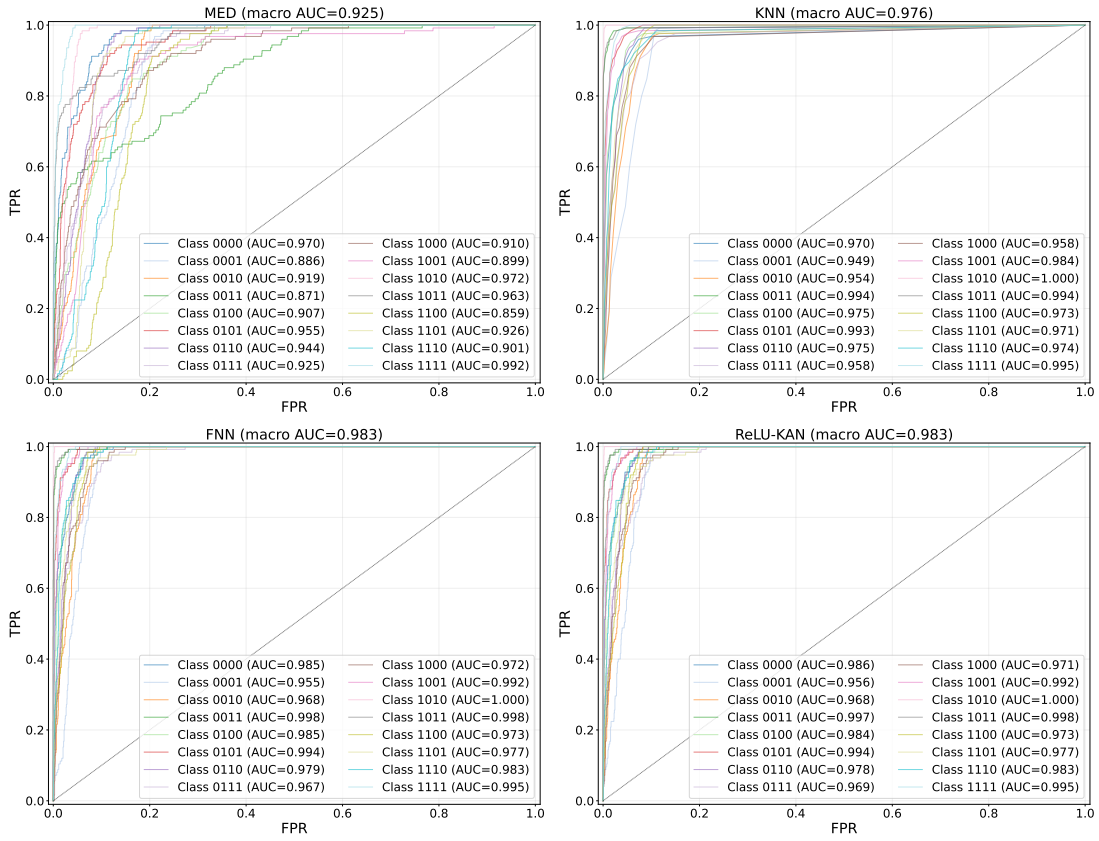


Figure 29 – One-versus-rest ROC curves for 16-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

with the FNN and ReLU-KAN sharing the highest value of 0.9826 for 16-CSK and the FNN leading slightly for 8-CSK at 0.9890. For 4-CSK (Fig. 27), all three learned classifiers achieve a macro AUC of 1.0000, indicating essentially perfect separability at the operating-point noise.

Decision boundaries.

Figs. 30–32 complement the preceding metrics by visualizing the Principal Component Analysis (PCA)-projected decision regions at $\sigma_{\mathrm{n}} = 5.9$ mV for each constellation order. The projection lays bare the root cause of the performance gap: the practical class clusters do not conform to a fixed Voronoi tessellation, and the learned models sculpt their boundaries to that warped geometry far more effectively than MED. The effect is subtlest for 4-CSK, where even the MED partitions capture most of the well-separated structure, and most pronounced for 16-CSK, where the Voronoi cells assigned by MED systematically bisect several overlapping clusters.

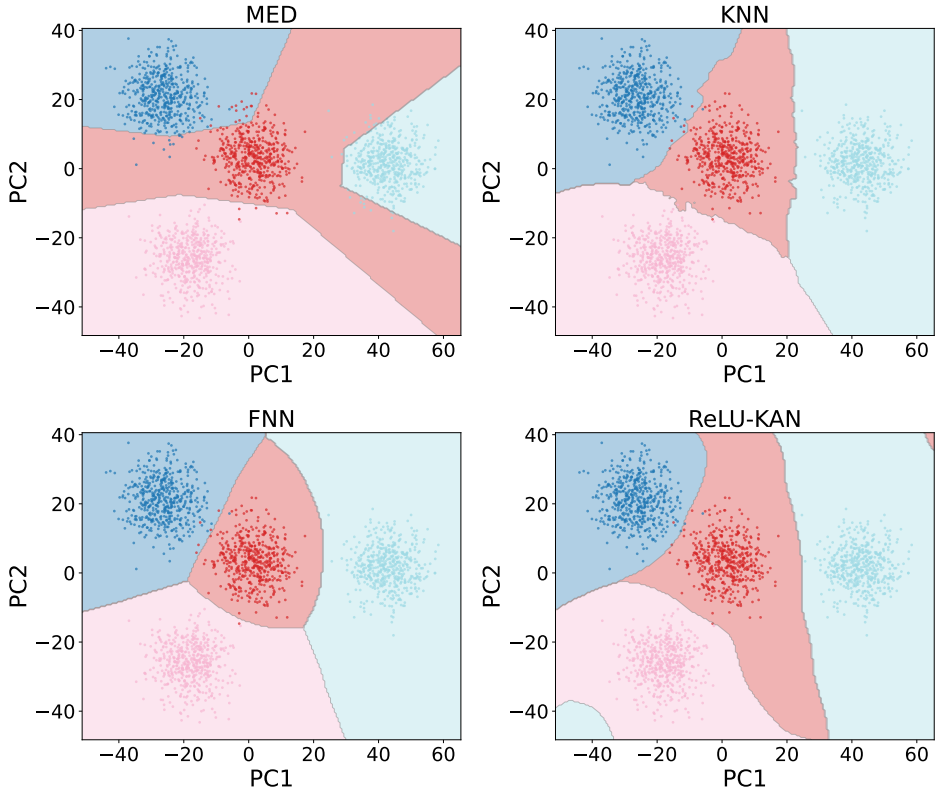


Figure 30 – PCA-projected decision regions for 4-CSK at $\sigma_{\mathrm{n}} = 5.9$ mV.

4.4.5 Sample Efficiency

Fig. 33 shows the classification accuracy as a function of the number of training samples per symbol. Each classifier is trained independently at each training-set size and evaluated on a fixed test set of 1 000 samples per symbol.

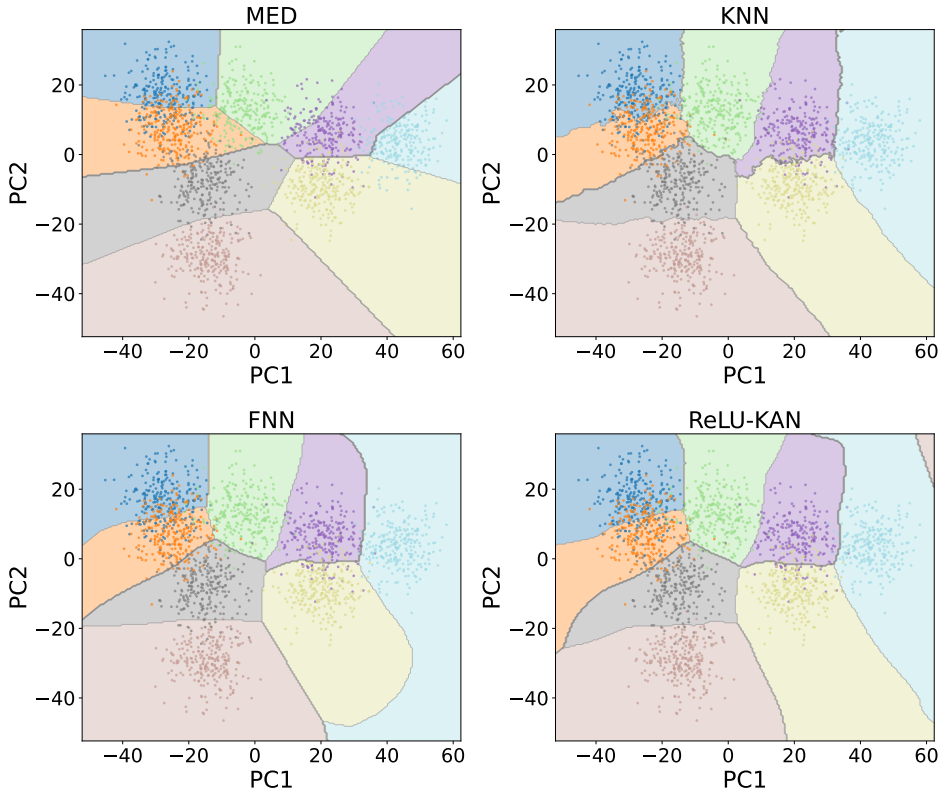


Figure 31 – PCA-projected decision regions for 8-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

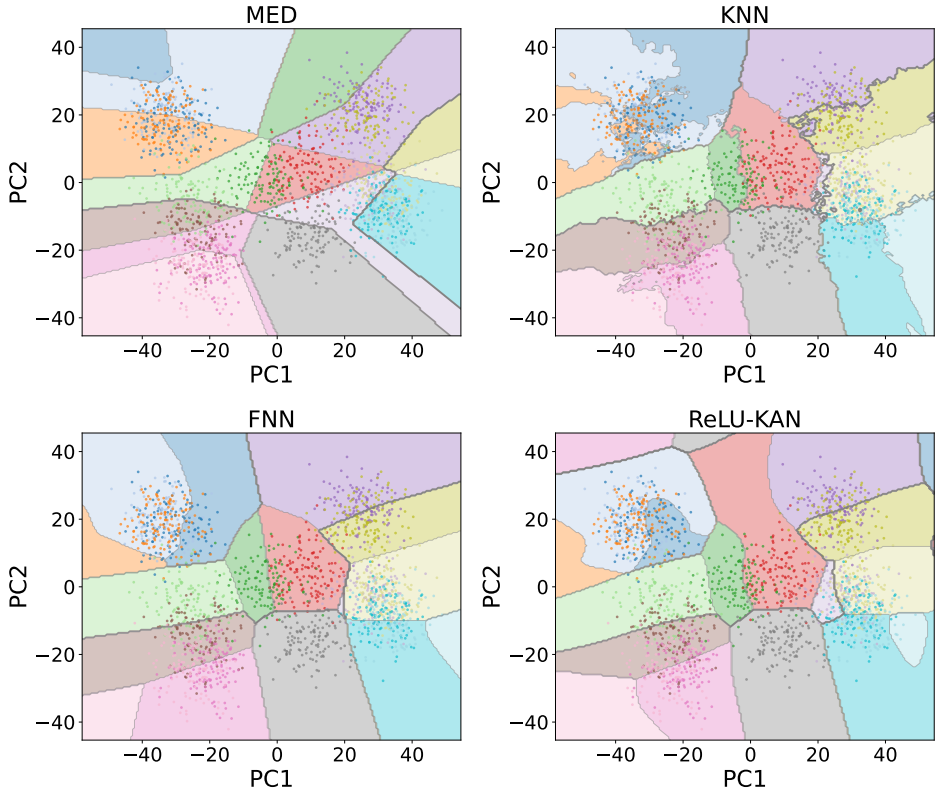
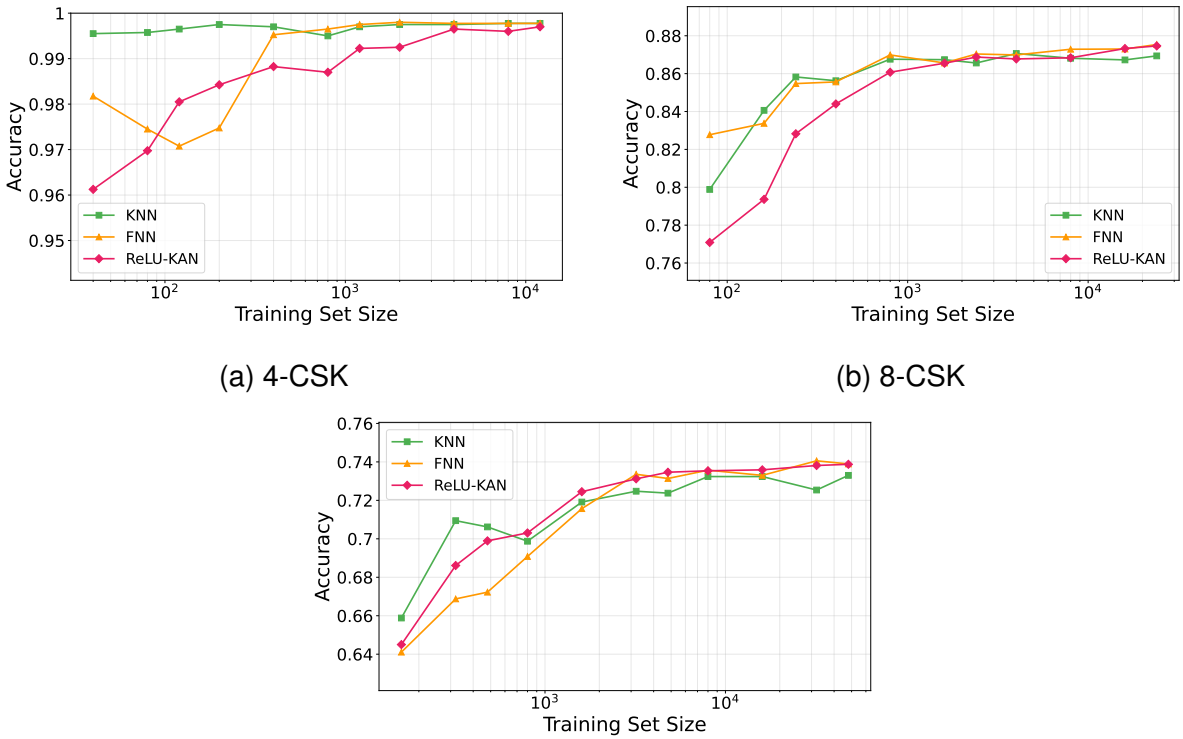


Figure 32 – PCA-projected decision regions for 16-CSK at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.



(a) 4-CSK

(b) 8-CSK

(c) 16-CSK

Figure 33 – Classification accuracy versus number of training samples per symbol for (a) 4-CSK, (b) 8-CSK, and (c) 16-CSK, at the operating-point noise $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$.

For 4-CSK, all learned classifiers converge rapidly: with only 10 training samples per symbol (40 total), kNN already attains 99.6% accuracy, while the FNN reaches 98.2% on the same budget and crosses the 99% mark once 100 samples per symbol (400 total) become available. The MED, being purely model-based, derives no benefit from additional data and remains flat at 95.2%. ReLU-KAN starts lower at 96.1% with 10 samples per symbol, but narrows the gap steadily and surpasses 98% once roughly 50 samples per symbol are supplied.

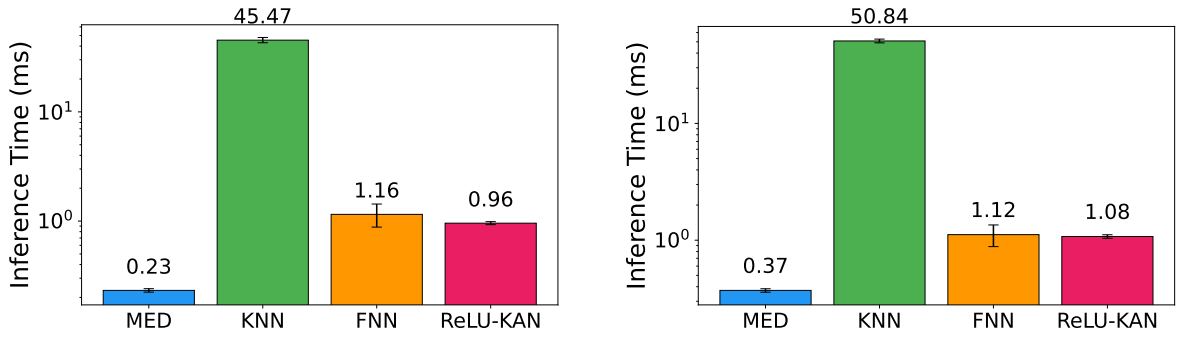
For the denser constellations, the learning curves settle into the characteristic plateau of diminishing returns beyond a few hundred samples per symbol. For 8-CSK, the learned classifiers reach accuracies of approximately 87% at the largest budget evaluated (3 000 samples per symbol), with FNN at 87.5%, ReLU-KAN at 87.5%, and kNN at 86.9%. The corresponding 16-CSK figures at 3 000 samples per symbol are 74.1% (FNN), 73.9% (ReLU-KAN), and 73.3% (kNN), compared to a flat MED baseline of 43.4%. These tightly clustered values confirm that all three learned classifiers converge to near-identical accuracy given sufficient data; the meaningful differentiator is sample efficiency at small training budgets, where kNN remains particularly competitive.

4.4.6 Computational Complexity

Table 19 and Fig. 34 summarize the measured computation times at the operating-point noise $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$ for all three constellation orders. Training uses 10 000 samples; the prediction time corresponds to the entire 2 000-sample test batch.

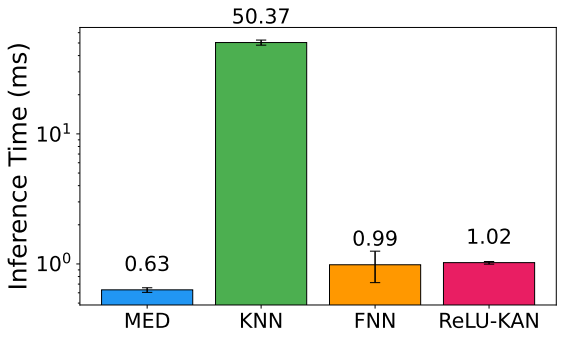
Table 19 – Training and inference time at $\sigma_{\mathrm{n}} = 5.9\,\mathrm{mV}$ for each constellation order. Best result per column pair in **bold**.

	4-CSK		8-CSK		16-CSK	
Classifier	Train (s)	Pred (ms)	Train (s)	Pred (ms)	Train (s)	Pred (ms)
MED	**0.0000**	**0.31**	**0.0000**	**0.42**	**0.0000**	**0.68**
KNN	0.0100	45.40	0.0096	50.72	0.0094	49.47
FNN	14.74	1.22	14.23	1.18	12.12	1.12
ReLU-KAN	13.32	1.18	13.43	1.13	13.39	1.23



(a) 4-CSK

(b) 8-CSK



(c) 16-CSK

Figure 34 – Comparison of training and prediction times for (a) 4-CSK, (b) 8-CSK, and (c) 16-CSK. Notice the inverse trade-off between the instantaneous training of KNN and the rapid inference of the neural architectures.

The MED requires no training time across all constellation orders, while completing predictions in sub-millisecond time. kNN likewise trains almost instantaneously, yet its prediction time, around 45–51 ms regardless of constellation order, is two orders of magnitude larger because every incoming sample must be compared against the entire stored training set. The neural models exhibit the inverse trade-off: offline

training requires 12–15 s depending on order and architecture, but once the weights are frozen, per-batch inference completes in roughly one millisecond. Notably, the ReLU-KAN training time remains stable at approximately 13.3–13.4 s across all orders, whereas the FNN varies more widely from 12.1 s for 16-CSK to 14.7 s for 4-CSK. Both neural architectures achieve comparable inference latency, with the FNN marginally faster at around 1.1–1.2 ms and ReLU-KAN at 1.1–1.2 ms per batch. This asymmetry between training and inference cost is especially attractive for deployment scenarios in which the detector is fitted once, for instance during system commissioning or periodic recalibration, and subsequently reused over many transmitted symbols without retraining. Furthermore, the inference latency of the FNN and ReLU-KAN models can be drastically reduced by leveraging parallel computing. Such parallelism is readily achievable through reconfigurable logic devices like Field-Programmable Gate Arrays (FPGAs), which are already widely utilized in modern communication systems.

4.5 Conclusion

This chapter established an end-to-end practical model for IEEE 802.15.7 CSK with OTS RGB optical components and used it to quantify how hardware nonidealities affect symbol detection. The analysis showed that the practical link departs from the ideal geometric model at several points: measured LED spectra do not reproduce all nominal IEEE chromaticity coordinates exactly; spectral overlap among RGB emitters and non-ideal optical filters induces channel coupling; wavelength-dependent photodiode responsivity changes the relative electrical gains of the three branches; and linear equalization mitigates but does not fully remove the resulting constellation warping. The noise model further connected the simulation parameter σ_{n} to a physical front-end reference: under the adopted parameters, the per-channel noise is dominated by shot noise, with a worst-case red-channel value near 4.3 mV and a three-branch root-sum-of-squares value of approximately 5.9 mV.

The numerical results demonstrate that the principal limitation of conventional MED is not only additive noise, but the mismatch between the nominal IEEE decision geometry and the practical received constellation. This mismatch is particularly severe for 16-CSK: even at $\sigma_{\mathrm{n}} = 1$ mV, where noise is nearly negligible, MED reaches only 70.9% accuracy. The learned detectors recover much of this deterministic loss by estimating decision regions directly from distorted receiver observations, reaching above 98% accuracy in the same near-pristine 16-CSK case. At the physically motivated operating point $\sigma_{\mathrm{n}} = 5.9$ mV, the advantage remains substantial: for 16-CSK, MED attains 43.95% accuracy and a macro-F1 score of 0.4159, whereas the best learned detector reaches 72.65% accuracy and a macro-F1 score of 0.7204. Similar, although less dramatic, gains occur for 8-CSK, and all methods approach near-perfect performance for

4-CSK.

The comparison among learned detectors indicates that the dominant benefit comes from adapting the decision rule to the distorted class geometry. The specific classifier choice is therefore a second-order design variable that depends on the deployment constraint. kNN is highly sample-efficient and essentially training-free, which is attractive for rapid calibration, but its inference cost scales with the stored dataset. The FNN and ReLU-KAN require offline training but provide millisecond-level batch inference in the reported implementation and are more suitable for parallel hardware acceleration. ReLU-KAN is particularly competitive in low-to-moderate noise for the denser constellations, whereas the FNN is the most robust option for high-noise 16-CSK in the evaluated operating range.

From a deployment perspective, the learned receiver should be interpreted as a calibrated detector matched to a defined operating envelope, rather than as a universally valid classifier for arbitrary optical front-ends. Its generalization depends on the extent to which the training procedure captures the relevant variations in LED spectra, dimming level, link geometry, receiver filtering, photodiode responsivity, equalizer state, and ambient-light noise. A detector trained under a given condition remains reliable while the received cluster centroids, spreads, and correlations remain sufficiently close to those observed during calibration. This requirement is relatively mild for sparse constellations such as 4-CSK, but becomes increasingly important for 8- and 16-CSK, where the inter-class margins are smaller and practical spectral distortion already places several clusters close to each other. In practical implementations, this limitation can be addressed by a pilot-based training procedure, which allows the receiver to periodically recalibrate or fine-tune the decision regions as the optical front-end, channel, or noise conditions vary.

The results also clarify why conventional transmitter-side nonlinear pre-distortion, although possible, does not eliminate the need for receiver-side compensation. Pre-distortion can modify the RGB drive currents and partially compensate current-to-radiant-flux nonlinearities in a controllable transmitter, but it cannot change the underlying spectral power distributions of the practical OTS LEDs. In CSK, this is a fundamental limitation: the transmitter can only reweight the available red, green, and blue spectral components, and cannot synthesize spectral content that the diodes do not physically emit. Therefore, when the nominal IEEE 802.15.7 chromaticity points are outside the physically attainable RGB gamut, or when broadband and overlapping spectra warp the constellation after color matching, filtering, propagation, and photodetection, transmitter-side pre-distortion can only reduce the deterministic bias within the feasible color region. The remaining mismatch is a physical feasibility error associated with the finite spectral gamut of the OTS source, not merely a digital implementation error.

Receiver-side learned classification is therefore well suited to this scenario because it learns the actually observed constellation after the combined effects of the transmitter, channel, receiver optics, photodetection, and noise. It also preserves a simple, reliable, low-cost transmitter architecture by avoiding additional lookup tables, nonlinear current-control loops, and high-resolution drive electronics. Thus, transmitter pre-distortion and learned detection should be viewed as complementary: pre-distortion may reduce deterministic centroid displacement when the transmitter is fully controllable, whereas the proposed receiver-side approach provides adaptive compensation while maintaining compatibility with simple OTS-based VLC hardware.

5 CONCLUSION

This dissertation investigated intelligent physical layer architectures for next-generation wireless communication systems. Three complementary problems were formulated and addressed, each targeting a distinct layer of the physical layer design pipeline: predictive reconfigurable surface control, joint sensing–communication beamforming optimization, and optical modulation detection under hardware nonidealities. Across all three contributions, the central finding is that incorporating learning-based or model-driven intelligence into well-established physical layer building blocks yields substantial and, in several cases, order-of-magnitude improvements with respect to conventional model-based approaches.

The remainder of this chapter first summarizes the specific conclusions drawn from each technical chapter and then presents the general conclusions and perspectives that emerge from the dissertation as a whole.

5.1 Specific Conclusions

5.1.1 Predictive RIS Control via Kalman Filtering

Chapter 2 addressed the temporal mismatch between the slow rate of GNSS position updates and the fast reconfiguration requirements of RIS-aided communications. A predictive control strategy based on the Kalman filter was proposed, employing both CV and CA motion models to anticipate user equipment trajectories and proactively adjust the RIS phase configuration between consecutive positioning measurements.

The numerical results demonstrated that predictive tracking substantially outperforms the naïve baseline strategy across all tested trajectory profiles and sampling rates. The most pronounced gains were observed in the low-frequency regime: at a GNSS update rate of 1 Hz, the CA Kalman filter reduced the average received-power degradation from -1.59 dB (naïve) to -0.33 dB, corresponding to an approximately fivefold improvement. This confirms that model-based state prediction can effectively bridge the temporal gap between sparse positioning measurements and real-time beam-steering requirements. The comparison between motion models further revealed that the CA model exhibits greater robustness in trajectories involving turning behavior or abrupt acceleration changes, while the CV model remains adequate for smoother, approximately rectilinear motion.

Despite these promising results, the study also identified important limitations.

The filter performance depends sensitively on the tuning of the process-noise and measurement-noise covariance matrices, and the adopted block-diagonal covariance structure may be inadequate for scenarios with strong inter-axis coupling. Moreover, the simplified path-loss propagation model and the single-user scope indicate that significant work remains before the approach can be extended to realistic multi-user deployments with complex channel conditions.

5.1.2 Deep Unfolded SCA for ISAC Beamforming

Chapter 3 tackled the computational bottleneck of non-convex beamforming optimization in ISAC systems, where the sensing and communication objectives share the same transmit waveform and must be jointly optimized under strict power and per-user SINR constraints. A model-driven deep unfolding architecture was proposed in which the classical SCA algorithm is reformulated as a fixed-depth computational graph, with learnable layer-dependent step sizes, momentum coefficients, and penalty weights replacing the static heuristic hyperparameters of the original solver.

The key finding is that the principal limitation of conventional iterative solvers in high-interference and latency-constrained settings is not asymptotic performance, but rather their inability to reach high-quality feasible solutions within a restricted computational budget. In the baseline loaded scenario ($K = 10$ users), the classical SCA solver achieved feasible solutions in only 11.3% of the tested channel realizations, whereas the proposed DU-SCA framework reached an 89.5% feasibility rate under the same iteration budget, representing a nearly eightfold improvement. This acceleration translated into sensing gains on the order of 1.25 to 1.75 dB in CRB, depending on the operating point. Analysis of the learned parameter trajectories revealed interpretable optimization strategies: the network learns to enforce power constraints aggressively only in later layers and to adopt a coarse-to-fine step-size schedule, behaviors that a manual tuning procedure would be unlikely to discover.

Nonetheless, the approach is fundamentally bounded by the feasibility limits of the underlying system configuration. In highly loaded regimes ($K \geq 14$), even the unfolded architecture exhibits residual constraint violations, and the training-distribution dependence implies that retraining may be necessary when the channel statistics change substantially.

5.1.3 Machine-Learning-Based Detection for IEEE 802.15.7 CSK

Chapter 4 investigated the detection problem in IEEE 802.15.7 CSK modulation when practical off-the-shelf optical components are employed. An end-to-end system model was developed, encompassing broadband LED spectra, non-ideal optical filters, wavelength-dependent photodiode responsivity, and the resulting coupled 3×3 optical–

electrical channel. Four detection strategies were compared: the conventional MED, kNN, a feedforward neural network (FNN), and the recently proposed ReLU-KAN.

The most significant finding is that the principal performance limiter in practical CSK is not additive noise but the geometric mismatch between the received constellation and the nominal IEEE reference geometry. This is strikingly demonstrated in the near-noiseless 16-CSK case ($\sigma_n = 1$ mV), where the MED achieves only 70.9% accuracy due to constellation warping alone, while all learned detectors exceed 98% accuracy. At the adopted operating point ($\sigma_n = 5.9$ mV, matching the root-sum-of-squares combined noise floor across the three receiver branches), the best learned detector improves 16-CSK accuracy from 43.95% to 72.65%, reducing the classification error by approximately 51%. The noise sweep revealed a nuanced ranking: for the well-separated 4-CSK constellation, kNN reaches perfect accuracy at the lowest noise levels while the FNN leads at high noise ($\sigma_n = 10$ mV); for 8-CSK, the three learned detectors perform comparably, with ReLU-KAN holding a slight edge at both intermediate and high noise; for the denser 16-CSK constellation, ReLU-KAN leads at low noise whereas the FNN becomes the most consistent performer from the operating point onward. Nevertheless, the gaps among learned classifiers remain small relative to their collective advantage over the geometric baseline.

The chapter also established practical deployment guidelines. kNN is the most sample-efficient detector, already attaining 99.6% accuracy for 4-CSK with only 10 samples per symbol, making it well suited for rapid initial calibration. The FNN and ReLU-KAN, on the other hand, require a one-time training investment of roughly 12–15 seconds but deliver batch inference on the order of one millisecond, making them attractive for real-time deployment.

5.2 General Conclusions

Taken together, the three contributions of this dissertation share a common theme: in each case, a well-understood physical layer component, beam steering, beamforming optimization, or symbol detection, encounters a performance ceiling that arises not from fundamental information-theoretic limits but from practical constraints such as sparse measurements, limited computational budgets, or hardware nonidealities. The introduction of learning-based or model-driven intelligence consistently lifts these ceilings by adapting to the specific operating conditions that conventional fixed-rule approaches cannot accommodate.

Several cross-cutting observations emerge from this work:

1. **Model-driven approaches preserve interpretability.** Rather than replacing established physical layer algorithms with opaque neural networks, the strategies

developed in Chapters 2 and 3 augment classical model-based methods, Kalman filtering and successive convex approximation, with learned components. This design philosophy yields solutions whose internal behavior can be analyzed and whose failure modes can be diagnosed, an important property for safety-critical communication systems.

2. **Data-driven detection compensates for hardware imperfections.** Chapter 4 demonstrated that supervised classifiers can absorb complex, nonlinear distortions introduced by practical optical hardware, recovering performance that would otherwise require expensive narrowband components. This finding has broader implications for any modulation scheme deployed with cost-constrained, off-the-shelf front ends.

3. **The dominant bottleneck is often not noise.** Across the three studies, the primary performance limiter was consistently a structural mismatch, temporal mismatch between measurement and control rates, computational mismatch between solver convergence speed and latency budgets, or geometric mismatch between ideal and practical constellation shape, rather than additive noise. Recognizing and addressing these mismatches through intelligent adaptation is, arguably, the most impactful design insight of this dissertation.

4. **Modest learning effort yields large gains.** The Kalman filter requires no training data; the deep unfolded optimizer trains on 1 000 channel realizations; and the CSK classifiers saturate with approximately 300 labeled samples per symbol. In all cases, the learning overhead is small compared to the performance improvement it enables, which supports the practical deployment of these techniques.

5. **Limitations remain at the boundaries of the operating envelope.** The predictive RIS strategy degrades under severe model mismatch; the unfolded beamformer cannot overcome fundamental feasibility limits in overloaded regimes; and the learned CSK detectors have not been validated under distribution shift across noise levels. These residual limitations delineate the boundary between what intelligent physical layer design can achieve within existing system architectures and what requires a more fundamental redesign.

In summary, this dissertation demonstrated that incorporating intelligence into the physical layer, whether through model-based prediction, algorithm unrolling, or data-driven classification, yields consistent, substantial, and practically realizable gains across diverse wireless communication scenarios relevant to future 6G networks. The results support the broader vision that the physical layer of next-generation systems will increasingly rely on learned components to bridge the gap between theoretical performance and practical deployment constraints.

REFERENCES

ARMIJO, Larry. Minimization of functions having Lipschitz continuous first partial derivatives. **Pacific Journal of Mathematics**, v. 16, n. 1, p. 1–3, 1966. DOI: 10.2140/pjm.1966.16.1.

BALATSOUKAS-STIMMING, Alexios; STUDER, Christoph. Deep Unfolding for Communications Systems: A Survey and Some New Directions. *In:* 2019 IEEE International Workshop on Signal Processing Systems (SiPS). [*S. l.: s. n.*], 2019. p. 266–271. DOI: 10.1109/SiPS47522.2019.9020494.

BECKER, Alex. **Kalman Filter from the Ground Up**. [*S. l.: s. n.*], 2023. ISBN 9789655984392.

BENGIO, Yoshua. Practical recommendations for gradient-based training of deep architectures. *In:* NEURAL Networks: Tricks of the Trade. [*S. l.*]: Springer, 2012. p. 437–478.

BERTSEKAS, D.P. **Nonlinear Programming**. [*S. l.*]: Athena Scientific, 1999.

BOYD, Stephen; VANDENBERGHE, Lieven. **Convex Optimization**. [*S. l.*]: Cambridge University Press, Mar. 2004. ISBN 9780511804441. DOI: 10.1017/cbo9780511804441. Available from: http://dx.doi.org/10.1017/CBO9780511804441.

BROWN, Robert Grover; HWANG, Patrick Y. C. **Introduction to Random Signals and Applied Kalman Filtering**. 3. ed. [*S. l.*]: John Wiley & Sons, 1996. ISBN 9780471128397.

CHEN, Zhen; CHEN, Gaojie; TANG, Jie; ZHANG, Shun; SO, Daniel K. C.; DOBRE, Octavia A.; WONG, Kai-Kit; CHAMBERS, Jonathon. Reconfigurable-Intelligent-Surface-Assisted B5G/6G Wireless Communications: Challenges, Solution, and Future Opportunities. **IEEE Communications Magazine**, v. 61, n. 1, p. 16–22, 2023. DOI: 10.1109/MCOM.002.2200047.

CHENG, Ziyang; LIAO, Bin; SHI, Shengnan; HE, Zishu; LI, Jun. Co-Design for Overlaid MIMO Radar and Downlink MISO Communication Systems via Cramér-Rao Bound Minimization. **IEEE Transactions on Signal Processing**, v. 67, n. 24, p. 6227–6240, 2019. DOI: 10.1109/TSP.2019.2952048.

CHOWDHURY, Mostafa Zaman; HOSSAN, Md. Tanvir; ISLAM, Amirul; JANG, Yeong Min. A Comparative Survey of Optical Wireless Technologies: Architectures and Applications. **IEEE Access**, v. 6, p. 9819–9840, 2018. DOI: 10.1109/ACCESS.2018.2792419.

COSTA, Filippo; BORGESE, Michele. Electromagnetic Model of Reflective Intelligent Surfaces. **IEEE Open Journal of the Communications Society**, v. 2, p. 1577–1589, 2021. DOI: 10.1109/OJCOMS.2021.3092217.

DEKA, Sukanya; DEKA, Kuntal; NGUYEN, Nhan Thanh; SHARMA, Sanjeev; BHATIA, Vimal; RAJATHEVA, Nandana. Comprehensive Review of Deep Unfolding Techniques for Next-Generation Wireless Communication Systems. **arXiv preprint arXiv:2502.05952**, 2025. DOI: 10.48550/arXiv.2502.05952. Available from: https://arxiv.org/abs/2502.05952.

DIMITROV, Svilen; HAAS, Harald. **Principles of LED Light Communications: Towards Networked Li-Fi**. Cambridge: Cambridge University Press, 2015. DOI: 10.1017/CBO9781107278929.

ELMOSSALLAMY, Mohamed A.; ZHANG, Hongliang; SONG, Lingyang; SEDDIK, Karim G.; HAN, Zhu; LI, Geoffrey Ye. Reconfigurable Intelligent Surfaces for Wireless Communications: Principles, Challenges, and Opportunities. **IEEE Transactions on Cognitive Communications and Networking**, v. 6, n. 3, p. 990–1002, 2020. DOI: 10.1109/TCCN.2020.2992604.

FABIANI, Mattia; SILVA, Diego; ABDALLAH, Asmaa; CELIK, Abdulkadir; ELTAWIL, Ahmed M. Multi-Modal Sensing and Communication for V2V Beam Tracking via Camera and GPS Fusion. *In:* 2024 58th Asilomar Conference on Signals, Systems, and Computers. [*S. l.: s. n.*], 2024. p. 1–6. DOI: 10.1109/IEEECONF60004.2024.10942842.

FAIRMAN, Hugh S.; BRILL, Michael H.; HEMMENDINGER, Henry. How the CIE 1931 color-matching functions were derived from Wright-Guild data. **Color Research & Application**, v. 22, n. 1, p. 11–23, 1997. DOI: https://doi.org/10.1002/(SICI)1520-6378(199702)22:1<11::AID-COL4>3.0.CO;2-7. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291520-6378%28199702%2922%3A1%3C11%3A%3AAID-COL4%3E3.0.CO%3B2-7. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291520-6378%28199702%2922%3A1%3C11%3A%3AAID-COL4%3E3.0.CO%3B2-7.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern Recognition Letters**, v. 27, n. 8, p. 861–874, June 2006. ISSN 0167-8655. DOI: 10.1016/j.patrec.2005.10.010. Available from: http://dx.doi.org/10.1016/j.patrec.2005.10.010.

GAO, Qian; ZHONG, Ruikang; ZOU, Yixuan; SHIN, Hyundong; LIU, Yuanwei. Deep Learning-Based Beamforming Optimization for ISAC Systems: A Low-Complexity and Transferable Framework. **IEEE Transactions on Wireless Communications**, v. 25, p. 8754–8768, 2026. DOI: 10.1109/TWC.2025.3649260.

GENG, Zuhang; KHAN, Faisal Nadeem; GUAN, Xun; DONG, Yuhan. Advances in Visible Light Communication Technologies and Applications. **Photonics**, v. 9, n. 12,

2022. ISSN 2304-6732. DOI: 10.3390/photonics9120893. Available from: https://www.mdpi.com/2304-6732/9/12/893.

GOLUB, Gene H.; VAN LOAN, Charles F. **Matrix Computations**. 4th. Baltimore, MD, USA: Johns Hopkins University Press, 2013.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [*S. l.*]: MIT Press, 2016. http://www.deeplearningbook.org.

GREGOR, Karol; LECUN, Yann. Learning Fast Approximations of Sparse Coding. *In:* PROC. Int. Conf. Mach. Learn. (ICML). Haifa, Israel: [*s. n.*], 2010. p. 399–406.

GREWAL, Mohinder S.; ANDREWS, Angus P. **Kalman Filtering: Theory and Practice Using MATLAB**. 3. ed. [*S. l.*]: Wiley-IEEE Press, 2008. ISBN 9780470173664.

HAKEEM, Shimaa A. Abdel; HUSSEIN, Hanan H.; KIM, HyungWon. Vision and Research Directions of 6G Technologies and Applications. **Journal of King Saud University - Computer and Information Sciences**, v. 34, 6, Part A, p. 2419–2442, 2022. DOI: 10.1016/j.jksuci.2022.03.019. Available from: https://www.sciencedirect.com/science/article/pii/S1319157822001033.

HE, Hengtao; JIN, Shi; WEN, Chao-Kai; GAO, Feifei; LI, Geoffrey Ye; XU, Zongben. Model-Driven Deep Learning for Physical Layer Communications. **IEEE Wireless Communications**, IEEE, v. 26, n. 5, p. 77–83, 2019. DOI: 10.1109/MWC.2019.1800447.

HE, Jiguang; WYMEERSCH, Henk; SANGUANPUAK, Tachporn; SILVEN, Olli; JUNTTI, Markku. Adaptive Beamforming Design for mmWave RIS-Aided Joint Localization and Communication. *In:* 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). [*S. l.: s. n.*], 2020. p. 1–6. DOI: 10.1109/WCNCW48565.2020.9124848.

HE, Yinghui; CAI, Yunlong; MAO, Hao; YU, Guanding. RIS-Assisted Communication Radar Coexistence: Joint Beamforming Design and Analysis. **IEEE Journal on Selected Areas in Communications**, v. 40, n. 7, p. 2131–2145, 2022. DOI: 10.1109/JSAC.2022.3155507.

HE, Zhenyao; XU, Wei; SHEN, Hong; NG, Derrick Wing Kwan; ELDAR, Yonina C.; YOU, Xiaohu. Full-Duplex Communication for ISAC: Joint Beamforming and Power Optimization. **IEEE Journal on Selected Areas in Communications**, v. 41, n. 9, p. 2920–2936, 2023. DOI: 10.1109/JSAC.2023.3287540.

HERSHEY, John R.; LE ROUX, Jonathan; WENINGER, Felix. Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures. **arXiv preprint**

arXiv:1409.2574, 2014. DOI: 10.48550/arXiv.1409.2574. Available from: https://arxiv.org/abs/1409.2574.

HUA, Haocheng; XU, Jie; HAN, Tony Xiao. Optimal Transmit Beamforming for Integrated Sensing and Communication. **IEEE Transactions on Vehicular Technology**, v. 72, n. 8, p. 10588–10603, 2023. DOI: 10.1109/TVT.2023.3262513.

HUA, Haocheng; XU, Jie; HAN, Tony Xiao. Transmit Beamforming Optimization for Integrated Sensing and Communication. *In:* 2021 IEEE Global Communications Conference (GLOBECOM). [*S. l.: s. n.*], 2021. p. 1–6. DOI: 10.1109/GLOBECOM46510.2021.9685478.

HUANG, Chongwen; HU, Sha; ALEXANDROPOULOS, George C.; ZAPPONE, Alessio; YUEN, Chau; ZHANG, Rui; RENZO, Marco Di; DEBBAH, Merouane. Holographic MIMO Surfaces for 6G Wireless Networks: Opportunities, Challenges, and Trends. **IEEE Wireless Communications**, v. 27, n. 5, p. 118–125, 2020. DOI: 10.1109/MWC.001.1900534.

HULL, Christopher. **1Hz GPS Tracking Data**. [*S. l.*]: Mendeley Data, 2024. DOI: 10.17632/XT69CNWH56.3. Available from: https://data.mendeley.com/datasets/xt69cnwh56/3.

IEEE. IEEE Standard for Local and metropolitan area networks–Part 15.7: Short-Range Optical Wireless Communications. **IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011)**, p. 1–407, 2019. DOI: 10.1109/IEEESTD.2019.8697198.

JOVICIC, Aleksandar; LI, Junyi; RICHARDSON, Tom. Visible light communication: opportunities, challenges and the path to market. **IEEE Communications Magazine**, v. 51, n. 12, p. 26–32, 2013. DOI: 10.1109/MCOM.2013.6685754.

KAHN, J.M.; BARRY, J.R. Wireless infrared communications. **Proceedings of the IEEE**, v. 85, n. 2, p. 265–298, 1997. DOI: 10.1109/5.554222.

KAPLAN, Elliott; HEGARTY, Christopher. [*S. l.: s. n.*], 2017.

KINGMA, Diederik P.; BA, Jimmy. Adam: A Method for Stochastic Optimization. *In:* INTERNATIONAL Conference on Learning Representations (ICLR). [*S. l.: s. n.*], 2015.

LI, Boning; VERMA, Gunjan; SEGARRA, Santiago. Graph-Based Algorithm Unfolding for Energy-Aware Power Allocation in Wireless Networks. **IEEE Transactions on Wireless Communications**, v. 22, n. 2, p. 1359–1373, 2023. DOI: 10.1109/TWC.2022.3204486.

LIU, An; HUANG, Zhe; LI, Min; WAN, Yubo; LI, Wenrui; HAN, Tony Xiao; LIU, Chenchen; DU, Rui; TAN, Danny Kai Pin; LU, Jianmin; SHEN, Yuan; COLONE, Fabiola; CHETTY, Kevin. A Survey on Fundamental Limits of Integrated

Sensing and Communication. **IEEE Communications Surveys & Tutorials**, v. 24, n. 2, p. 994–1034, 2022. DOI: 10.1109/COMST.2022.3149272.

LIU, An; LAU, Vincent K. N.; KANANIAN, Borna. Stochastic Successive Convex Approximation for Non-Convex Constrained Stochastic Optimization. **IEEE Transactions on Signal Processing**, v. 67, n. 16, p. 4189–4203, 2019. DOI: 10.1109/TSP.2019.2925601.

LIU, Fan; CUI, Yuanhao; MASOUROS, Christos; XU, Jie; HAN, Tony Xiao; ELDAR, Yonina C.; BUZZI, Stefano. Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond. **IEEE Journal on Selected Areas in Communications**, v. 40, n. 6, p. 1728–1767, 2022. DOI: 10.1109/JSAC.2022.3156632.

LIU, Fan; LIU, Ya-Feng; LI, Ang; MASOUROS, Christos; ELDAR, Yonina C. Cramér-Rao Bound Optimization for Joint Radar-Communication Beamforming. **IEEE Transactions on Signal Processing**, v. 70, p. 240–253, 2022. DOI: 10.1109/TSP.2021.3135692.

LIU, Fan; LIU, Ya-Feng; MASOUROS, Christos; LI, Ang; ELDAR, Yonina C. A Joint Radar-Communication Precoding Design Based on Cramér-Rao Bound Optimization. *In:* 2022 IEEE Radar Conference (RadarConf22). [*S. l.: s. n.*], 2022. p. 1–6. DOI: 10.1109/RadarConf2248738.2022.9764187.

LIU, Fan; MASOUROS, Christos; LI, Ang; RATNARAJAH, Tharmalingam. Robust MIMO Beamforming for Cellular and Radar Coexistence. **IEEE Wireless Communications Letters**, v. 6, n. 3, p. 374–377, 2017. DOI: 10.1109/LWC.2017.2693985.

LIU, Fan; MASOUROS, Christos; PETROPULU, Athina P.; GRIFFITHS, Hugh; HANZO, Lajos. Joint Radar and Communication Design: Applications, State-of-the-Art, and the Road Ahead. **IEEE Transactions on Communications**, v. 68, n. 6, p. 3834–3862, 2020. DOI: 10.1109/TCOMM.2020.2973976.

LIU, Fan; ZHOU, Longfei; MASOUROS, Christos; LI, Ang; LUO, Wu; PETROPULU, Athina. Toward Dual-functional Radar-Communication Systems: Optimal Waveform Design. **IEEE Transactions on Signal Processing**, v. 66, n. 16, p. 4264–4279, 2018. DOI: 10.1109/TSP.2018.2847648.

LIU, Yanzhen; HU, Qiyu; CAI, Yunlong; YU, Guanding; LI, Geoffrey Ye. Deep-Unfolding Beamforming for Intelligent Reflecting Surface Assisted Full-Duplex Systems. **IEEE Transactions on Wireless Communications**, v. 21, n. 7, p. 4784–4800, 2022. DOI: 10.1109/TWC.2021.3132924.

LIU, Yuanwei; LIU, Xiao; MU, Xidong; HOU, Tianwei; XU, Jiaqi; RENZO, Marco Di; AL-DHAHIR, Naofal. Reconfigurable Intelligent Surfaces: Principles and

Opportunities. **IEEE Communications Surveys & Tutorials**, v. 23, n. 3, p. 1546–1577, 2021. DOI: 10.1109/COMST.2021.3077737.

LIU, Ziming; WANG, Yixuan; VAIDYA, Sachin; RUEHLE, Fabian; HALVERSON, James; SOLJAČIĆ, Marin; HOU, Thomas Y.; TEGMARK, Max. **KAN: Kolmogorov-Arnold Networks**. [*S. l.: s. n.*], 2025. arXiv: 2404.19756 [cs.LG]. Available from: https://arxiv.org/abs/2404.19756.

MARIOTTO, Flávio Tonioli; YOMA, Néstor Becerra; ALMEIDA, Madson Cortes de. Unsupervised Competitive Learning Clustering and Visual Method to Obtain Accurate Trajectories From Noisy Repetitive GPS Data. **IEEE Transactions on Intelligent Transportation Systems**, v. 26, n. 2, p. 1562–1572, 2025. DOI: 10.1109/TITS.2024.3520393.

MATHIAS, Luis Carlos; DE MELO, Leonimer Flávio; ABRAO, Taufik. Modeling and mitigation of spectral crosstalk in OFDM WDM-VLC system. **Optics Commun.**, v. 478, p. 126361, 2021. DOI: https://doi.org/10.1016/j.optcom.2020.126361.

MATHIAS, Luis Carlos; SOUZA, Alvaro Ricieri Castro e; ABRÃO, Taufik. Wavelength widths of optical filters for optimum SINR in WDM-VLC systems. **Appl. Opt.**, v. 59, p. 5615–5624, 2020. Available from: https://opg.optica.org/ao/abstract.cfm?URI=ao-59-18-5615.

MONGA, Vishal; LI, Yuelong; ELDAR, Yonina C. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. **IEEE Signal Processing Magazine**, v. 38, n. 2, p. 18–44, 2021. DOI: 10.1109/MSP.2020.3016905.

NGUYEN, Nhan Thanh; NGUYEN, Ly V.; SHLEZINGER, Nir; ELDAR, Yonina C.; SWINDLEHURST, A. Lee; JUNTTI, Markku. Joint Communications and Sensing Hybrid Beamforming Design via Deep Unfolding. **IEEE Journal of Selected Topics in Signal Processing**, v. 18, n. 5, p. 901–916, 2024. DOI: 10.1109/JSTSP.2024.3463403.

NGUYEN, Nhan Thanh; SHLEZINGER, Nir; ELDAR, Yonina C.; JUNTTI, Markku. Multiuser MIMO Wideband Joint Communications and Sensing System With Subcarrier Allocation. **IEEE Transactions on Signal Processing**, v. 71, p. 2997–3013, 2023. DOI: 10.1109/TSP.2023.3302622.

NOCEDAL, Jorge; WRIGHT, Stephen J. **Numerical Optimization**. 2nd. New York, NY, USA: Springer, 2006.

ONYEKPE, Uche; PALADE, Vasile; KANARACHOS, Stratis; SZKOLNIK, Alicja. IO-VNBD: Inertial and Odometry Benchmark Dataset for Ground Vehicle Positioning. **Data in Brief**, v. 35, p. 106885, 2021. DOI: 10.1016/j.dib.2021.106885. Available from: https://www.sciencedirect.com/science/article/pii/S2352340921001694.

PARIKH, Neal; BOYD, Stephen. Proximal Algorithms. **Foundations and Trends in Optimization**, Now Publishers, v. 1, n. 3, p. 127–239, 2014.

PASCANU, Razvan; MIKOLOV, Tomas; BENGIO, Yoshua. On the difficulty of training recurrent neural networks. *In:* INTERNATIONAL Conference on Machine Learning (ICML). [*S. l.: s. n.*], 2013. p. 1310–1318.

PATHAK, Parth H.; FENG, Xiaotao; HU, Pengfei; MOHAPATRA, Prasant. Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges. **IEEE Communications Surveys & Tutorials**, v. 17, n. 4, p. 2047–2077, 2015. DOI: 10.1109/COMST.2015.2476474.

PAUL, Bryan; CHIRIYATH, Alex R.; BLISS, Daniel W. Survey of RF Communications and Sensing Convergence Research. **IEEE Access**, v. 5, p. 252–270, 2017. DOI: 10.1109/ACCESS.2016.2639038.

QIU, Qi; ZHU, Tao; GONG, Helin; CHEN, Liming; NING, Huansheng. ReLU-KAN: New Kolmogorov-Arnold Networks that Only Need Matrix Addition, Dot Multiplication, and ReLU. *In:* 2025 IEEE Smart World Congress (SWC). [*S. l.: s. n.*], 2025. p. 1686–1694. DOI: 10.1109/SWC65939.2025.00262.

RUDER, Sebastian. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016.

SAAD, Walid; BENNIS, Mehdi; CHEN, Mingzhe. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. **IEEE Network**, v. 34, n. 3, p. 134–142, 2020. DOI: 10.1109/MNET.001.1900287.

SÄRKKÄ, Simo. **Bayesian Filtering and Smoothing**. Cambridge: Cambridge University Press, 2013.

SCUTARI, Gesualdo; FACCHINEI, Francisco; SONG, Peiran; PALOMAR, Daniel P.; PANG, Jong-Shi. Decomposition by Partial Linearization: Parallel Optimization of Multi-Agent Systems. **IEEE Transactions on Signal Processing**, v. 62, n. 3, p. 641–656, 2014. DOI: 10.1109/TSP.2013.2293126.

SHI, Qingjiang; HONG, Mingyi. Penalty Dual Decomposition Method for Nonsmooth Nonconvex Optimization—Part I: Algorithms and Convergence Analysis. **IEEE Transactions on Signal Processing**, v. 68, p. 4108–4122, 2020. DOI: 10.1109/TSP.2020.3001906.

SHI, Qingjiang; HONG, Mingyi; FU, Xiao; CHANG, Tsung-Hui. Penalty Dual Decomposition Method for Nonsmooth Nonconvex Optimization—Part II: Applications. **IEEE Transactions on Signal Processing**, v. 68, p. 4242–4257, 2020. DOI: 10.1109/TSP.2020.3001397.

SINGH, Rohit; KAUSHIK, Aryan; SHIN, Wonjae; RENZO, Marco Di; SCIANCALEPORE, Vincenzo; LEE, Doohwan; SASAKI, Hirofumi; SHOJAEIFARD, Arman; DOBRE, Octavia A. Toward 6G Evolution: Three Enhancements, Three Innovations, and Three Major Challenges. **IEEE Network**, v. 39, n. 6, p. 139–147, 2025. DOI: 10.1109/MNET.2025.3574717.

SUN, Ying; BABU, Prabhu; PALOMAR, Daniel P. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. **IEEE Transactions on Signal Processing**, v. 65, n. 3, p. 794–816, 2017. DOI: 10.1109/TSP.2016.2601299.

SUTSKEVER, Ilya; MARTENS, James; DAHL, George; HINTON, Geoffrey. On the importance of initialization and momentum in deep learning. *In:* INTERNATIONAL Conference on Machine Learning (ICML). [*S. l.: s. n.*], 2013. p. 1139–1147.

SUTTON, Richard S. Adapting bias by gradient descent: an incremental version of delta-bar-delta. *In:* PROCEEDINGS of the Tenth National Conference on Artificial Intelligence. San Jose, California: AAAI Press, 1992. (AAAI'92), p. 171–176. ISBN 0262510634.

TEMIZ, Murat; ZHANG, Yongwei; FU, Yanwei; ZHANG, Chi; MENG, Chenfeng; KAPLAN, Orhan; MASOUROS, Christos. Deep-Learning-Based Techniques for Integrated Sensing and Communication Systems: State-of-the-Art, Challenges, and Opportunities. **IEEE Open Journal of the Communications Society**, v. 6, p. 5940–5968, 2025. DOI: 10.1109/OJCOMS.2025.3586560.

TONG, Chang-Dong; ZHENG, Xi; HUANG, Xiao; ZENG, Pei-Xin; DENG, Quan; CHEN, Ya-Yong; WU, Ting-Zhu; LU, Yi-Jun; CHEN, Zhong; GUO, Wei-Jie. The Impacts of Optical Crosstalk on the Color Gamut of Mini-LED Integrated Matrix Device. **IEEE Photon. Technol. Lett.**, v. 36, p. 957–960, 2024.

U-BLOX. **ZED-F9P-04B High Precision GNSS Module Data Sheet**. [*S. l.*], 2022. Document No. UBX-21044850. Available from: https://content.u-blox.com/sites/default/files/ZED-F9P-04B_DataSheet_UBX-21044850.pdf.

WANG, Cheng-Xiang; YOU, Xiaohu; GAO, Xiqi; ZHU, Xiuming; LI, Zixin; ZHANG, Chuan; WANG, Haiming; HUANG, Yongming; CHEN, Yunfei; HAAS, Harald; THOMPSON, John S.; LARSSON, Erik G.; RENZO, Marco Di; TONG, Wen; ZHU, Peiying; SHEN, Xuemin; POOR, H. Vincent; HANZO, Lajos. On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds. **IEEE Communications Surveys & Tutorials**, v. 25, n. 2, p. 905–974, 2023. DOI: 10.1109/COMST.2023.3249835.

WANG, Zhaocheng; WANG, Qi; HUANG, Wei; XU, Zhengyuan. **Visible light communications: modulation and signal processing**. Hoboken: Wiley-IEEE Press, 2017. (IEEE series on digital & mobile communication). ISBN 9781119331858.

WU, Qingqing; ZHANG, Rui. Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network. **IEEE Communications Magazine**, v. 58, n. 1, p. 106–112, 2020. DOI: 10.1109/MCOM.001.1900107.

WU, Qingqing; ZHANG, Shuowen; ZHENG, Beixiong; YOU, Changsheng; ZHANG, Rui. Intelligent Reflecting Surface-Aided Wireless Communications: A Tutorial. **IEEE Transactions on Communications**, v. 69, n. 5, p. 3313–3351, 2021. DOI: 10.1109/TCOMM.2021.3051897.

ZHANG, Jianjun; MASOUROS, Christos; LIU, Fan; HUANG, Yongming; SWINDLEHURST, A. Lee. Low-Complexity Joint Radar-Communication Beamforming: From Optimization to Deep Unfolding. **IEEE Journal of Selected Topics in Signal Processing**, v. 19, n. 5, p. 856–870, 2025. DOI: 10.1109/JSTSP.2024.3522787.

ZHANG, Jifa; LIU, Mingqian; TANG, Jie; ZHAO, Nan; NIYATO, Dusit; WANG, Xianbin. Joint Design for RIS-Aided ISAC via Deep Unfolding Learning. **IEEE Transactions on Cognitive Communications and Networking**, v. 11, n. 1, p. 349–362, 2025. DOI: 10.1109/TCCN.2024.3435422.

ZHANG, Jifa; ZHU, Yongxu; ZHAO, Nan; JIN, Shi; WANG, Xianbin; NG, Derrick Wing Kwan; AL-DHAHIR, Naofal. Deep Unfolding Learning Aided ISAC Transceiver Design. **IEEE Transactions on Wireless Communications**, v. 24, n. 10, p. 8681–8696, 2025. DOI: 10.1109/TWC.2025.3429384.

ZHANG, Zhengquan; XIAO, Yue; MA, Zheng; XIAO, Ming; DING, Zhiguo; LEI, Xianfu; KARAGIANNIDIS, George K.; FAN, Pingzhi. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. **IEEE Vehicular Technology Magazine**, v. 14, n. 3, p. 28–41, 2019. DOI: 10.1109/MVT.2019.2921208.

ZHANG, Zhongze; JIANG, Tao; YU, Wei. Active Sensing for Localization With Reconfigurable Intelligent Surface. *In:* ICC 2023 - IEEE International Conference on Communications. [*S. l.: s. n.*], 2023. p. 4261–4266. DOI: 10.1109/ICC45041.2023.10279094.

APPENDIX

APPENDIX A – PREDICTIVE CONTROL FOR RIS-AIDED B5G NETWORKS USING KALMAN FILTERS

Predictive Control for RIS-aided B5G Networks using Kalman Filters

Conference Paper

Accepted and presented in the **XLIII Brazilian Symposium on Telecommunications and Signal Processing (SBrT 2025)**, September 29th to October 2nd, Natal, RN.

Submission Date	16 Jul. 2025
Status	Accepted and presented

Predictive control for RIS-aided B5G networks using Kalman filters

Rafael Marasca Martins, Luis Carlos Mathias, and Taufik Abrão

***Abstract*— Reconfigurable Intelligent Surfaces (RIS) are a key enabler for B5G/6G wireless networks, particularly in dense urban environments, enhancing spectral efficiency through controlled signal propagation. However, accurate User Equipment (UE) tracking is essential for real-time RIS reconfiguration, especially in scenarios where the system depends on low-rate, noisy Global Positioning System (GPS) updates. To address this, we propose a Kalman filter-based control algorithm that estimates the UE's position between GPS samples, allowing the RIS to optimize received power. Results show that this method can boost received power fivefold, demonstrating its effectiveness as a low-complexity solution for real-time beamforming in mobile scenarios with sparse location data.**

***Keywords*— RIS, Kalman filter, Predictive control, B5G, 6G.**

I. Introduction

One of the greatest challenges in modern communications within dense urban environments is overcoming multipath fading, which arises from signal reflections and attenuation caused by interactions with the surrounding environment, such as building foundations, metal structures, and other obstacles [1], [2]. In this context, Intelligent Reflecting Surfaces (RIS) emerge as a promising enabling technology, as they can make the propagation environment controllable by passively and electronically adjusting the phase of reflected signals [1]–[3]. This capability allows RIS to create constructive or destructive interference in specific spatial regions, enhancing signal quality where needed. Another important advantage of RIS is its ability to mitigate shadowing by creating a virtual Line-of-Sight (LoS) between the Base Station (BS) and the User Equipment (UE) [1], [3].

However, a key requirement of Beyond 5G (B5G) and 6G networks is Ultra-Reliable Low-Latency Communication (URLLC) [3], [4]. This imposes strict constraints on the control algorithms, which must operate with minimal latency to reduce the overhead during the RIS reconfiguration phase—particularly the communication delays between the UE, BS, and RIS. Therefore, minimizing this reconfiguration overhead is critical for ensuring high Quality of Service (QoS), especially for mobile users [2], [3].

However, when a precise and high-rate sensing infrastructure is not available, the adjustment of the RIS must rely on low-rate and noisy Global Positioning System (GPS) measurements of the UE's position. But, as the wireless communication channel is intrinsically stochastic and highly volatile, mainly due to factors such as weather changes, user mobility, and the dynamic presence of obstacles [1], the RIS is unable to accurately reconfigure itself to track the UE's position adequately, resulting in degraded beam alignment and reduced communication performance [2]. This highlights the need for predictive control strategies capable of estimating the UE's trajectory in real time, even with sparse and uncertain measurements.

Several works, such as [5] and [6], explore Integrated Sensing and Communications (ISAC) techniques; however, their applicability can be limited by power, throughput, or interference constraints [7]. As a complementary direction, [8] proposes a transformer-based method that fuses camera and GPS data to predict beam states up to 500 ms in advance, enhancing received power. Nevertheless, this approach relies heavily on large volumes of data, which may not always be practical. Similarly, [9] presents an unsupervised clustering algorithm to estimate the true vehicle trajectory from sparse and noisy GPS data. While effective, these machine learning–based, data-centric solutions depend on substantial training data, which may limit their deployment in real-world scenarios.

A promising solution lies in the use of the Kalman filter. By leveraging past observations and a motion model, the Kalman filter provides optimal estimates of the system state, making it particularly well-suited for real-time tracking and prediction in dynamic environments such as mobile wireless networks [10]. In this context, the present work proposes a predictive control algorithm based on Kalman filtering, which uses low-rate GPS measurements from a mobile UE to estimate its position between samples. These predictions enable proactive RIS reconfiguration that compensates for control latency and anticipates the UE's movement, ultimately improving received signal power. This approach aims to maintain high communication performance while avoiding power loss penalties of reactive reconfiguration.

II. Kalman Filter

The Kalman filter is a recursive algorithm widely used for estimating the state of dynamic systems. It operates in two main steps: prediction and update [10], [11].

The prediction step estimates the current system state $\mathbf{s}_i$ and its associated uncertainty based on the previous state, assuming a linear system model with additive Gaussian process noise. The underlying system dynamics are described by [10]–[12]:

$$\mathbf{s}_i = \mathbf{F}\mathbf{s}_{i-1} + \mathbf{w}_{i-1}, \quad \mathbf{w}_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{1}$$

where $\mathbf{s}_i$ is the state at time step i, $\mathbf{F}$ is the state transition matrix, and $\mathbf{w}_{i-1}$ is the process noise with covariance matrix

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

R. Martins and T. Abrão are with the Department of Electrical Engineering, State University of Londrina (UEL), Londrina, PR, Brazil.

L. Mathias is with the Department of Computer Engineering, Federal University of Technology – Parana (UTFPR), PR, Brazil.

$\mathbf{Q}$. Based on this model, the predicted state estimate $\hat{\mathbf{s}}_i$ and its associated error covariance matrix $\widehat{\mathbf{P}}_i$ is computed as follows [10]–[12]:

$$\widehat{\mathbf{s}}_i = \mathbf{F}\mathbf{s}_{i-1} \tag{2}$$

$$\widehat{\mathbf{P}}_i = \mathbf{F}\mathbf{P}_{i-1}\mathbf{F}^T + \mathbf{Q} \tag{3}$$

where the notation $\widehat{(\cdot)}$ stands for the predicted estimate before the measurement update, and ^{T} represents the transpose of a vector or matrix. The update step is triggered when a new measurement is available. This step refines the predicted state using new information. It starts by computing the innovation covariance matrix $\mathbf{S}$ [10]–[12]:

$$\mathbf{S} = \mathbf{H}\mathbf{P}_i\mathbf{H}^T + \mathbf{R} \tag{4}$$

where $\mathbf{H}$ is the observation matrix that maps the predicted state to the measurement space, and $\mathbf{R}$ is the measurement noise covariance matrix, representing sensor uncertainties.

Next, the Kalman gain acts as a weighting factor that determines the relative importance of the measurement versus the prediction. It is denoted by $\mathbf{G}$ and can be calculated as follows [10]–[12]:

$$\mathbf{G} = \widehat{\mathbf{P}}_i\mathbf{H}^T\mathbf{S}^{-1} \tag{5}$$

The residual vector $\mathbf{e}_i$, representing the difference between the actual UE position measurement $\mathbf{r}_i$ and the predicted measurement, is given by [11]:

$$\mathbf{e}_i = \mathbf{r}_i - \mathbf{H}\widehat{\mathbf{s}}_i \tag{6}$$

The actual state and its covariance are then updated as follows [10]–[12]:

$$\mathbf{s}_i = \widehat{\mathbf{s}}_i + \mathbf{G}\mathbf{e}_i \tag{7}$$

$$\mathbf{P}_i = (\mathbf{I} - \mathbf{G}\mathbf{H})\widehat{\mathbf{P}}_i \tag{8}$$

A. System Model

In this paper, two motion models are considered to describe the system dynamics: a constant acceleration (CA) model and a constant velocity (CV) model. Accordingly, the state vectors for each model are defined as follows:

$$\mathbf{s}_{\text{CA}} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T \tag{9}$$

$$\mathbf{s}_{\text{CV}} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T \tag{10}$$

where x, y, and z denote the coordinates of the UE; $\dot{x}$, $\dot{y}$, and $\dot{z}$ denote the corresponding velocity components; and $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ represent the acceleration components (included only in the CA model).

The Kalman filter's state vector is initialized using the first available GPS measurement, under the assumption that the UE starts from rest. That is, all initial velocities and accelerations are set to zero at the beginning of the estimation process.

Based on the classical kinematic equations of motion, the system state can be extrapolated over a time interval Δt under the CV or CA assumption. These equations form the basis for the state transition models employed in this work, which are derived based on [10], [12].

Constant Acceleration Model: In the CA model, the position, the corresponding state transition matrix $\mathbf{F}_{\text{CA}}$ for the full 9-dimensional state vector is:

$$\mathbf{F}_{\text{CA}} = \mathbf{I_3} \otimes \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where $\otimes$ denotes the Kroenecker product.

Constant Velocity Model: In the CV model, the system assumes no acceleration; therefore, the position changes linearly with velocity, and velocity remains constant. Thus, the corresponding 6-dimensional state transition matrix $\mathbf{F}_{\text{CV}}$ is [12]:

$$\mathbf{F}_{\text{CV}} = \mathbf{I_3} \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{12}$$

These transition matrices are used in the prediction step of the Kalman filter to estimate the next state based on the current state and the underlying motion model.

B. Covariance Matrices

Assuming the estimation errors across the x, y, and z axes are uncorrelated, the initial error covariance matrix $\mathbf{P}$ and process noise covariance matrix $\mathbf{Q}$ are structured in a block-diagonal form.

Error Covariance Matrix: The error covariance matrix $\mathbf{P}$ follows the general definition [11], [12]:

$$\mathbf{P} = \mathbb{E}\{\mathbf{s}\mathbf{s}^T\} \tag{13}$$

For both the CV and CA models, $\mathbf{P}$ is block-diagonal per axis:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_z \end{bmatrix} \tag{14}$$

Here, $\mathbf{0}$ denotes a zero matrix of appropriate dimensions, and each $\mathbf{P}_k$ ($k \in \{x, y, z\}$) is a square matrix of size $q \times q$, where $q = 2$ for the CV model and $q = 3$ for the CA model:

$$\mathbf{P}_k = \begin{cases} \begin{bmatrix} p_{kk} & p_{k\dot{k}} \\ p_{\dot{k}k} & p_{\dot{k}\dot{k}} \end{bmatrix}, & \text{CV model} \\ \begin{bmatrix} p_{kk} & p_{k\dot{k}} & p_{k\ddot{k}} \\ p_{\dot{k}k} & p_{\dot{k}\dot{k}} & p_{\dot{k}\ddot{k}} \\ p_{\ddot{k}k} & p_{\ddot{k}\dot{k}} & p_{\ddot{k}\ddot{k}} \end{bmatrix}, & \text{CA model} \end{cases}$$

The initial covariance matrix is typically initialized with high variances to reflect uncertainty, especially in velocity and acceleration states.

Process Noise Covariance Matrix: The process noise covariance matrix $\mathbf{Q}$ captures the uncertainty introduced by the system dynamics and is derived by projecting the continuous-time noise model $\mathbf{Q}_c$ through the system transition matrix $\mathbf{F}$ [11]:

$$\mathbf{Q} = \mathbf{F}\mathbf{Q}_c\mathbf{F}^\top \tag{15}$$

In the constant acceleration model, the process noise is typically attributed to random perturbations in acceleration, which propagate into both velocity and position. In contrast, the constant velocity model assumes that only the velocity component is affected by noise.

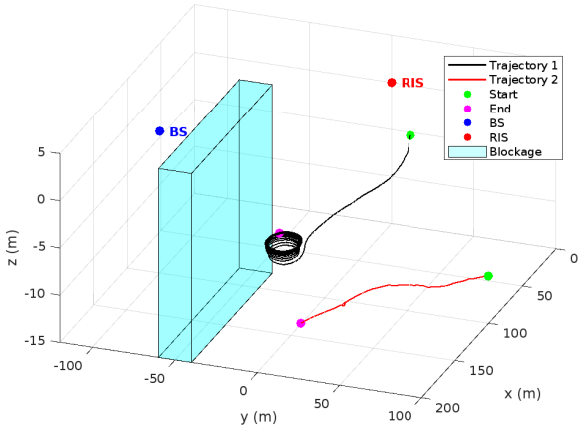


Fig. 1. Trajectories of the UE extracted from real datasets

C. Measurement Model

In the proposed system, each measurement corresponds to the UE's Cartesian position relative to the RIS. Therefore, the measurement matrix $\mathbf{H}$ maps the full state vector to the observed position components. It effectively selects the position entries (x, y, z) from the state vector while ignoring the unobserved velocity and acceleration states.

Assuming uncorrelated errors across dimensions, the measurement noise covariance matrix $\mathbf{R}$ is:

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \tag{16}$$

where σ_x^2, σ_y^2, and σ_z^2 denote the measurement variances along each axis.

In the remainder of this paper, we adopt the CA model as the default state-space formulation for the Kalman filter. Therefore, unless otherwise specified, all references to Kalman filter pertain to the CA model.

III. Methodology

The system considered in this paper comprises a fixed BS located at position $[100, -100, 0]$ m, a RIS at $[0, 0, 0]$ m, and a UE allowed to move freely within a defined area. We assume that, throughout the UE's trajectory, a physical obstacle obstructs the LoS path between the BS and the UE as depicted in Fig. 1. Both the UE and BS are equipped with isotropic antennas.

To simulate realistic vehicular motion, we employ two trajectories sourced from real-world GPS datasets. The first (Trajectory 1) is taken from [13], which contains 10 Hz GPS samples collected from various cars and drivers operating on public roads across the United Kingdom, Nigeria, and France. The second (Trajectory 2) comes from [14], featuring 1 Hz tracking data of South African minibus taxis, capturing dynamic driving behaviors such as aggressive acceleration. To enable quasi-continuous system evaluation, these signals are upsampled to 100 Hz via linear interpolation, which is sufficient given the relatively smooth nature of the underlying movements.

In addition to the empirical trajectories, a synthetic scenario (trajectory 3) is evaluated through a Monte Carlo simulation with 100 repetitions. Here, the UE follows a simulated path generated by a random-walk process in acceleration. At each time step, the position is updated using the kinematic equations:

$$\begin{cases} \mathbf{r}_t = \mathbf{r}_{t-1} + \dot{\mathbf{r}}_{t-1}\Delta t + 0.5\ddot{\mathbf{r}}\Delta t^2, \\ \dot{\mathbf{r}}_t = \dot{\mathbf{r}}_{t-1} + \ddot{\mathbf{r}}\Delta t \end{cases} \tag{17}$$

where the acceleration $\ddot{\mathbf{r}}_t$ at each time step is independently drawn from a Gaussian distribution with zero mean and variance 0.1 m/s^2. Both the initial position and velocity are assumed to be zero. This formulation enables a wide range of motion patterns.

The UE's trajectory is estimated in real time using a Kalman filter based on a constant acceleration motion model, and a Kalman filter based on the constant velocity model. These filters operate with a dual-rate structure: a high-frequency prediction step and a low-frequency update step. The prediction step is executed every 10 ms, producing rapid estimates of the UE's position, velocity, and acceleration, which are used to continuously adjust the RIS configuration for optimal signal reflection. Conversely, the update step occurs only when new GPS data is received, reflecting the sampling rate of the source. During this step, the filter refines its internal state by incorporating the new measurement, thereby correcting any accumulated drift.

For each predicted state, the phase shift that maximizes the received power at the UE can be determined for each RIS element as [15]:

$$\angle\Gamma_{m,n} = \mathrm{mod}\left(k_0\left(|r^{\mathrm{tx}}_{m,n}| + |\hat{r}^{\mathrm{rx}}_{m,n}|\right), 2\pi\right) \tag{18}$$

where $|r^{\mathrm{tx}}_{m,n}|$ and $|\hat{r}^{\mathrm{rx}}_{m,n}|$ are the exact distance from the BS and the estimated distance from the UE, respectively, to the (m,n)-th RIS element. If correctly estimated $|\hat{r}^{\mathrm{rx}}_{m,n}|$, the phase shifts obtained by (18) maximize the received power by allowing total constructive interference at the UE position.

The channel is modeled using a basic path-loss model, where the received power is given by Eq. 19 [15].

$$P_r = \frac{P_t\lambda^2}{(4\pi)^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{G_t(\theta^{\mathrm{tx}}_{m,n})G_r(\theta^{\mathrm{rx}}_{m,n})\sigma(\theta^t_{m,n},\theta^r_{m,n})|\Gamma_{m,n}|^2}{|r^{\mathrm{tx}}_{m,n}|^2|r^{\mathrm{rx}}_{m,n}|^2} \times e^{-2j\angle\Gamma_{m,n}+jk_0(|r^{\mathrm{tx}}_{m,n}|+|r^{\mathrm{rx}}_{m,n}|)} \tag{19}$$

In which P_t denotes the transmit power, λ is the signal wavelength, and G_t and G_r represent the transmit and receive antenna gains, respectively, both assumed to be 1 due to isotropic radiation. The RIS comprises M vertical and N horizontal elements. The angles $\theta^t_{m,n}$ and $\theta^r_{m,n}$ correspond to the angles of incidence and reflection at the (m,n)-th RIS element, while $|r^{\mathrm{rx}}_{m,n}|$ is the exact distance between the UE and the (m,n)-th RIS element. The radar cross-section of a unit cell, denoted by σ, is given by [15]:

$$\sigma(\theta^t_{m,n},\theta^r_{m,n}) = 4\pi\left(\frac{D^2}{\lambda}\right)^2\cos^2(\theta^t_{m,n}) \times\left(\frac{\sin\left(\frac{kD}{2}(\sin\theta^r_{m,n}-\sin\theta^t_{m,n})\right)}{\frac{kD}{2}(\sin\theta^r_{m,n}-\sin\theta^t_{m,n})}\right)^2 \tag{15}$$

where D is the periodicity of the elements, which is taken as $\lambda/2$ in this work.

IV. Numerical Results

This section compares the performance of the Kalman filter-based RIS configuration strategy with a naive approach, which reconfigures the RIS only when a new GPS coordinate is provided. The comparison is based on the received power at the UE, evaluated over time and under different sampling rates (1 Hz, 5 Hz, and 10 Hz). Figures 2, 3, and 4 illustrate the instantaneous power received at the UE for the three different employed trajectories (Fig. 4 represents the data from a single sample of the Monte Carlo simulation). Table I summarizes each condition's average power gain (in dB).

TABLE I
Comparison of Power Gain [dB] in UE for Different Methods and Sampling Rates for Three Trajectories

Trajectory	Method	Sampling Rate (Hz)		
		1Hz	5Hz	10Hz
Trajectory 1	Kalman	-0.3290	-0.3462	-0.0454
	Kalman (CV)	-0.3707	-0.2875	-0.2542
	Naive	-1.5868	-0.3471	-0.0841
Trajectory 2	Kalman	-0.3389	-0.0838	-0.0438
	Kalman (CV)	-0.3372	-0.1013	-0.0533
	Naive	-0.6102	-0.1239	-0.1070
Trajectory 3	Kalman	-0.2341	-0.0516	-0.0357
	Kalman (CV)	-0.2200	-0.0408	-0.0270
	Naive	-0.3429	-0.1615	-0.1644

The results clearly demonstrate that the Kalman filter provides superior performance compared to the naive method. For all trajectories and sampling rates, the Kalman-based approach leads to consistently lower power losses. Notably, for Trajectory 1 at 1 Hz, the Kalman filter reduces power loss by nearly 5 times compared to the naive method, highlighting the significant benefit of using prediction in scenarios with infrequent updates.

Another consistent trend across all results is the impact of the sampling rate. As the sampling frequency increases, the power loss decreases for both methods. This is expected, as more frequent reconfiguration allows the RIS to better adapt to the UE's motion. At lower sampling rates (particularly 1 Hz), both methods suffer due to the longer intervals between updates. However, the naive approach is more severely affected, since it fully relies on outdated information and does not attempt to estimate the UE's position.

In the case of the Kalman filter, the reduced performance at lower sampling rates arises from two factors: modeling error, as the motion dynamics are less accurately captured over longer prediction horizons, and information decay, since the filter's predictions become less reliable in the absence of new measurements. This highlights the importance of both accurate motion models and timely updates in predictive tracking.

A further insight emerges when comparing the constant velocity Kalman filter to the standard Kalman filter. In scenarios involving abrupt motion changes or non-linear trajectories, such as in Trajectory 1, where the UE follows a circular path, the CV model struggles to maintain accurate tracking. Because it assumes linear motion with no acceleration, it cannot adapt to changes in speed or direction, resulting in systematic tracking errors. Specifically, it tends to lag behind during turns and overshoot when the UE slows down or stops. These inaccuracies manifest as fluctuations in received power, especially at lower sampling rates, where the time between updates exacerbates the model's limitations. Overall, the standard Kalman filter exhibits more stable and reliable performance in dynamic scenarios.

V. Conclusions and Next Research Project

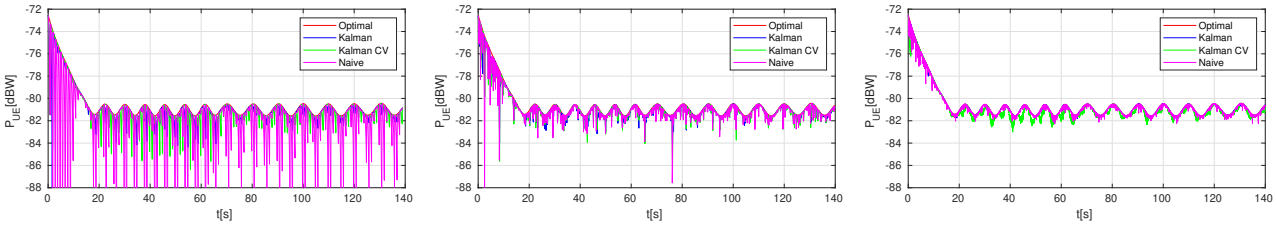
This paper presents a predictive RIS control strategy based on Kalman filtering to compensate for the limited availability of high-rate UE location data. In particular, the paper addresses scenarios where the UE position is obtained from low-rate GPS updates, a common constraint in real-world deployments.

Through extensive simulations over different trajectories and sampling rates, it is demonstrated that the Kalman filter can significantly improve RIS reconfiguration performance compared to a naive update strategy. The results show up to a 5-fold reduction in average power loss at the UE. The benefit becomes more pronounced as the sampling interval increases, highlighting the predictive capability of the Kalman filter in maintaining signal alignment over longer periods without fresh measurements.

This work also compared two motion models within the Kalman filter framework: the standard CA model and the CV model. The CV model exhibits reduced accuracy during turns or abrupt motion changes, as it does not account for acceleration. This limitation is particularly evident in Trajectory 1, where the UE follows a circular path.

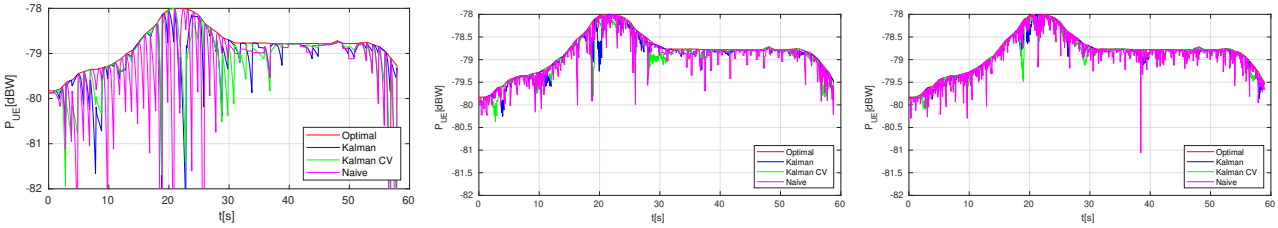
Notice that the Kalman filter requires careful tuning of its process and measurement noise covariance matrices to ensure satisfactory performance. Poorly chosen parameters can lead to unstable estimates or degraded tracking accuracy, especially under varying mobility conditions. Additionally, several other factors can introduce errors in Kalman filtering, including its reliance on accurate system modeling, the assumption of Gaussian noise, and the requirement for linear dynamics—all of which, when violated, can negatively impact estimation quality. Indeed, while simplifications made in modeling both the UE's movement (CV/CA models) and the radio environment (basic path-loss, isotropic antennas, ideal RIS) become the problem tractable and demonstrate the core concept effectively, they mean that the reported performance gains (e.g., "up to a fivefold improvement") should be seen as an optimistic upper bound. Real-world deployments will likely diminish gains due to more erratic UE motion, complex multipath, and hardware imperfections. Besides, the system's performance could degrade substantially due to severe GPS errors (outages, large outliers common in urban canyons) or practical RIS limitations (discrete phases, losses, switching times).

The focus on a single UE and specific trajectory types means that its direct applicability to dense, multi-user scenarios or highly diverse mobility patterns is not fully established. Hence, while the core predictive idea is sound, significant further research would be needed to scale this solution to the complexity of real B5G/6G deployments.



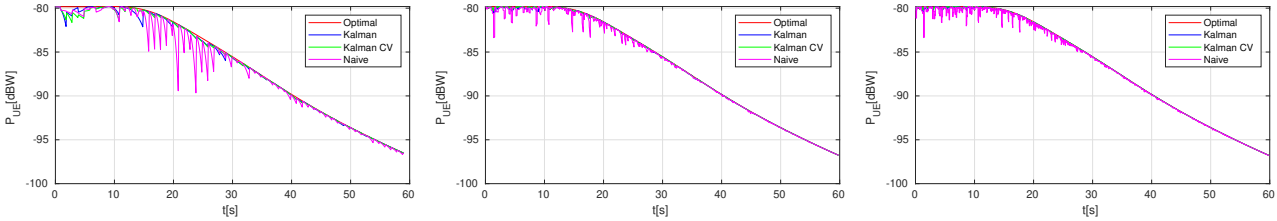
(a) 1Hz (b) 5Hz (c) 10Hz

Fig. 2. Power loss at the UE for trajectory 1 considering (a) 1Hz (b) 5Hz (c) 10Hz sampling rates



(a) 1Hz (b) 5Hz (c) 10Hz

Fig. 3. Power loss at the UE for trajectory 2 considering (a) 1Hz (b) 5Hz (c) 10Hz sampling rates



(a) 1Hz (b) 5Hz (c) 10Hz

Fig. 4. Power loss at the UE for trajectory 3 (simulated) considering (a) 1Hz (b) 5Hz (c) 10Hz sampling rates

Future work should focus on incorporating more realistic motion models and advanced state estimation techniques, such as the Interacting Multiple Model Kalman Filter (IMM-KF), which combines multiple motion models to improve tracking performance in systems with switching dynamics, or machine learning-based predictors. In parallel, more sophisticated scenarios, channel and RIS models should be considered to better reflect real-world propagation conditions.

REFERENCES

[1] M. A. ElMossallamy, H. Zhang, L. Song, K. G. Seddik, Z. Han, and G. Y. Li, "Reconfigurable intelligent surfaces for wireless communications: Principles, challenges, and opportunities," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 990–1002, 2020.

[2] Z. Chen, G. Chen, J. Tang, S. Zhang, D. K. So, O. A. Dobre, K.-K. Wong, and J. Chambers, "Reconfigurable-intelligent-surface-assisted B5G/6G wireless communications: Challenges, solution, and future opportunities," *IEEE Communications Magazine*, vol. 61, no. 1, pp. 16–22, 2023.

[3] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[4] S. A. Abdel Hakeem, H. H. Hussein, and H. Kim, "Vision and research directions of 6G technologies and applications," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part A, pp. 2419–2442, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157822001033

[5] J. He, H. Wymeersch, T. Sanguanpuak, O. Silven, and M. Juntti, "Adaptive beamforming design for mmWave RIS-aided joint localization and communication," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6.

[6] Z. Zhang, T. Jiang, and W. Yu, "Active sensing for localization with reconfigurable intelligent surface," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 4261–4266.

[7] Y. He, Y. Cai, H. Mao, and G. Yu, "RIS-assisted communication radar coexistence: Joint beamforming design and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 7, pp. 2131–2145, 2022.

[8] M. Fabiani, D. Silva, A. Abdallah, A. Celik, and A. M. Eltawil, "Multimodal sensing and communication for V2V beam tracking via camera and GPS fusion," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 2024, pp. 1–6.

[9] F. T. Mariotto, N. B. Yoma, and M. C. d. Almeida, "Unsupervised competitive learning clustering and visual method to obtain accurate trajectories from noisy repetitive GPS data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 2, pp. 1562–1572, 2025.

[10] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[11] A. P. A. Mohinder S. Grewal, *Kalman filtering: Theory and practice using MATLAB*, 3rd ed. Wiley-IEEE Press, 2008.

[12] P. Y. C. H. Robert Grover Brown, *Introduction to Random Signals and Applied Kalman Filtering, Third Edition*, 3rd ed. John Wiley & Sons, 1996.

[13] U. Onyekpe, V. Palade, S. Kanarachos, and A. Szkolnik, "IO-VNBD: Inertial and odometry benchmark dataset for ground vehicle positioning," *Data in Brief*, vol. 35, p. 106885, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340921001694

[14] C. Hull, "1Hz GPS tracking data," 2024. [Online]. Available: https://data.mendeley.com/datasets/xt69cnwh56/3

[15] F. Costa and M. Borgese, "Electromagnetic model of reflective intelligent surfaces," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1577–1589, 2021.

APPENDIX B – MITIGATING NON-LINEAR DISTORTIONS IN IEEE 802.15.7 CSK MODULATION

Mitigating Non-Linear Distortions in IEEE 802.15.7 CSK Modulation: From Classical to KAN Classifiers

Journal Article

Submitted to the **IEEE/Optica Journal of Lightwave Technology (JLT)** on January 16th, 2026.

Manuscript ID: JLT-39464-2026.

Impact Factor	4.7 (2024 JCR)
Qualis Capes	A1
Submission Date	16 Jan. 2026
Status	Accepted
DOI	10.1109/JLT.2026.3683593

Mitigating Non-Linear Distortions in IEEE 802.15.7 CSK Modulation: From Classical to KAN Classifiers

Rafael Marasca Martins, Luis Carlos Mathias, Pedro Henrique Horst, Kapila Palitharathna, Taufik Abrão *Senior Member, IEEE*, Ioannis Krikidis *Fellow, IEEE*

***Abstract*—In Visible Light Communication (VLC), Color Shift Keying (CSK) conveys symbols through the relative instantaneous intensities of multi-colored Light-Emitting Diodes (LEDs). IEEE Std 802.15.7 defines the transmitted CSK constellation, but its idealized formulation assumes monochromatic emitters and leaves receiver-side tristimulus processing open to implementation. Consequently, when CSK is implemented with off-the-shelf (OTS) LEDs, optical filters (OFs), and photodetectors (PDs), the received symbols no longer follow the ideal reference constellation, but are shifted by spectral mismatch and constellation distortion. This makes analytical detection more challenging and motivates a receiver model that adapts to the distorted decision regions. In this work, we therefore model such an OTS CSK-VLC link and cast symbol detection as a supervised classification problem using receiver features extracted with and without channel equalization. For symbol detection, we conducted a comprehensive comparative analysis of 10 machine learning (ML) classifiers, spanning classical statistical methods, feedforward neural networks (FNNs), and the novel Kolmogorov-Arnold Networks (KANs), against a minimum Euclidean distance (MED) detector used as a baseline. Our results reveal a clear performance hierarchy. Linear classifiers exhibit an error floor caused by channel nonlinearities, while conventional FNNs struggle to converge under limited training data. In contrast, the KAN architecture overcomes both limitations and consistently delivers the best detection performance. Overall, all evaluated ML classifiers surpass the analytical MED baseline, whose performance is degraded by the mismatch between the ideal reference constellation and the distorted received symbols. Among the tested methods, KAN achieves the lowest symbol error rate (SER), of 9.06% for 8-CSK, reducing the error of standard FNNs by 53.3% and by 45.7% for 4-CSK modulation.**

***Index Terms*—Color shift keying, visible light communication, off-the-shelf components, ML classifiers, Kolmogorov-Arnold networks.**

I. Introduction

THE growing number of mobile devices has intensified the demand for data connectivity, thereby increasing the burden on the already scarce radio frequency spectrum [1]. In this scenario, Visible Light Communication (VLC) has become a promising alternative, primarily because of its vast, unlicensed, and widely available frequency band, together with the extensive deployment of artificial lighting systems based on Light-Emitting Diodes (LEDs), which allow faster light modulation [1], [2]. Among VLC modulation schemes, Color Shift Keying (CSK) is particularly attractive because it conveys symbols through the relative intensities of different colored LEDs [3]. In addition, CSK preserves control over the emitted illumination color and is compatible with discrete drive levels that can be efficiently implemented in LED driver circuits.

CSK is adopted in the PHY III layer of IEEE Std 802.15.7 [4]. Although the standard defines the transmitted CSK signal format and its constellation in the CIE 1931 color space, receiver processing remains implementation-dependent. In practical implementations based on off-the-shelf (OTS) LEDs, optical filters (OFs), and photodetectors (PDs), the spectral spread of the LEDs and the mismatch among receiver components distort the received constellation relative to the ideal constellation [2]. As a result, the decision regions assumed by conventional detectors no longer accurately represent the received signal distribution, making reliable symbol detection a challenging task.

Motivated by these practical limitations, this work investigates CSK symbol detection from a machine learning (ML) perspective. Instead of relying exclusively on ideal-reference constellation geometry, the proposed approach treats detection as a data-driven classification problem and evaluates different feature representations and hyperparameter configurations under realistic receiver distortions. In this context, multiple ML classifiers are compared with a classical minimum Euclidean distance (MED) detector in order to assess their suitability for practical CSK reception.

A. Related Works

ML has been successfully applied to several VLC tasks, including channel estimation, equalization, symbol detection [5], and localization [6]. For CSK in particular, prior work demonstrates that data-driven classifiers can model nonlinear decision boundaries that are difficult to capture with fixed analytical receivers.

In camera-based CSK communication, Convolutional Neural Network (CNN)-based demodulators were shown to improve detection by exploiting spatial structure in the received

R. M. Martins and T. Abrão are with Electrical Engineering Department, Universidade Estadual de Londrina, Londrina-Paraná, 86057-970, Brazil. E-mails: rafael.marasca@uel.br; taufik@uel.br

L. C. Mathias and P. H. Horst are with the Coordination of the Computer Engineering Course, Federal University of Technology - Paraná, Toledo, Paraná, 85902-490, Brazil (e-mail: mathias@utfpr.edu.br; ilmthorst@gmail.com).

K. Palitharathna and I. Krikidis are with IRIDA – Research Centre for Communication Technologies, Department of Electrical and Computer Engineering, University of Cyprus. E-mails: palitharathna.kapila@ucy.ac.cy; krikidis.ioannis@ucy.ac.cy

Manuscript received January 15, 2026; revised xxxxx xx, 2026.

signal [7]. Neural equalization has also enabled high-order camera-based CSK demodulation (up to 512 colors) under significant distortion [8]. In addition, deep neural network (DNN)-based detectors were investigated in advanced VLC architectures with reconfigurable intelligent surface (RIS) assistance [9]. These studies establish the potential of learned detectors, but camera-based links are constrained by sampling rate and therefore do not directly address high-bandwidth receivers based on discrete photodetectors.

Recent underground VLC studies further highlight the importance of evaluating CSK under realistic operating conditions. The bit error rate (BER) analysis with imperfect channel state information (CSI) in mining tunnels [10] and the adaptive angle-diversity receiver design for 6G underground VLC [11] indicate that robust reception depends on adaptation to channel non-idealities. Likewise, wavelength-dependent underground channel modeling with CSK evaluation [12] and IEEE 802.15.7 performance analysis in underground mines [13] show that practical spectral and propagation effects can lead to behavior that departs from ideal standard assumptions. These observations are consistent with our focus on OTS-induced spectral mismatch and robust CSK symbol detection.

Accordingly, this work investigates CSK demodulation using OTS LEDs, OFs, and PDs under explicit spectral mismatch. Within a unified evaluation framework, 10 ML classifier families are compared against the MED benchmark, with particular emphasis on the impact of feature representation and hyperparameter selection on detection performance. To the best of the authors' knowledge, this is the first systematic comparison of ML-based symbol detectors for CSK systems implemented with realistic OTS optical components under spectrally distorted channel conditions.

B. Organization

The remainder of this paper is organized as follows. Section II presents the system model and defines the receiver features. Section III summarizes the classifier families under evaluation. Section IV reports the numerical results. Section V discusses these results. Finally, Section VI concludes the paper.

II. System Model

Fig. 1 summarizes the considered CSK link. The system is organized according to the signal path. In block I, the input bits are mapped to a transmitted chromaticity symbol $\mathbf{s} = [x\, y]^{\mathrm{T}}$ in the CIE 1931 chromaticity plane. In block II, this symbol is converted into the RGB drive currents $\mathbf{I} = [I_{\mathrm{R}}\, I_{\mathrm{G}}\, I_{\mathrm{B}}]^{\mathrm{T}}$ applied to the colored LEDs through a digital-to-analog converter (DAC) and current driver stage. The emitted optical signals propagate through the optical channel and are separated at the receiver by OFs. The PDs and transimpedance amplifiers (TIAs) generate the received voltage vector $\mathbf{V}$. In block III, channel equalization produces the equalized RGB intensity vector $\mathbf{V}_{\mathrm{CE}} = [R\, G\, B]^{\mathrm{T}} = \hat{\mathbf{H}}^{-1}\mathbf{V}$. In block IV, $\mathbf{V}_{\mathrm{CE}}$ is mapped to the CIE 1931 tristimulus vector $\mathbf{C}'_{\mathrm{CE}} = [X'\, Y'\, Z']^{\mathrm{T}}$ and then projected onto the chromaticity estimate $\mathbf{s}'_{\mathrm{CE}} = [x'\, y']^{\mathrm{T}}$. Finally, in block V, the transmitted symbol index is recovered from $\mathbf{s}'_{\mathrm{CE}}$.

Throughout the paper, the symbol $[.]^{\mathrm{T}}$ denotes the transpose of a vector or matrix. More details on each stage are given next.

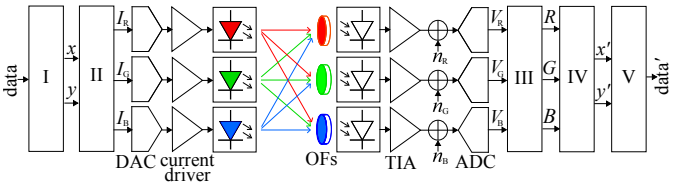


Fig. 1. Simplified VLC CSK system architecture.

A. Photometric and Chromaticity Preliminaries

The photometric representation adopted in this work follows the CIE 1931 colorimetric framework. For a spectral power distribution $S(\lambda)$, the corresponding CIE 1931 tristimulus vector is expressed as:

$$\mathbf{C} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \int_{\lambda_{\mathrm{L}}}^{\lambda_{\mathrm{H}}} S(\lambda)\bar{x}(\lambda)d\lambda \\ \int_{\lambda_{\mathrm{L}}}^{\lambda_{\mathrm{H}}} S(\lambda)\bar{y}(\lambda)d\lambda \\ \int_{\lambda_{\mathrm{L}}}^{\lambda_{\mathrm{H}}} S(\lambda)\bar{z}(\lambda)d\lambda \end{bmatrix}, \tag{1}$$

where $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are the CIE 1931 color matching functions (CMF), whereas λ_{L} and λ_{H} denote the lower and upper wavelength limits of the visible range, respectively.

The associated chromaticity coordinate vector is obtained by normalizing the tristimulus values as:

$$\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X}{X+Y+Z} \\ \frac{Y}{X+Y+Z} \end{bmatrix}. \tag{2}$$

Hence, Eqs. (1) and (2) provide the mapping from any optical spectrum to a point in the CIE 1931 chromaticity plane. Fig. 2 shows the CIE 1931 color matching functions together with the normalized spectral distributions of the colored OTS LEDs with a maximum nominal electrical power of 1 W considered in [2]. These photometric definitions are used throughout the remainder of the section to describe both symbol generation at the transmitter and chromaticity reconstruction at the receiver.

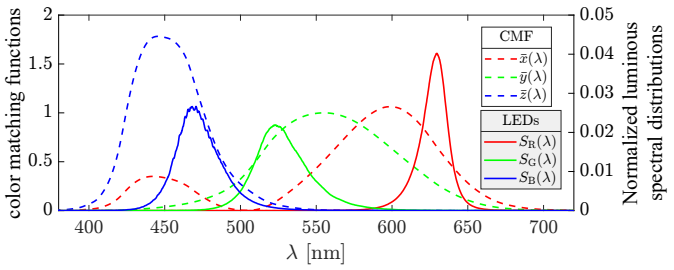
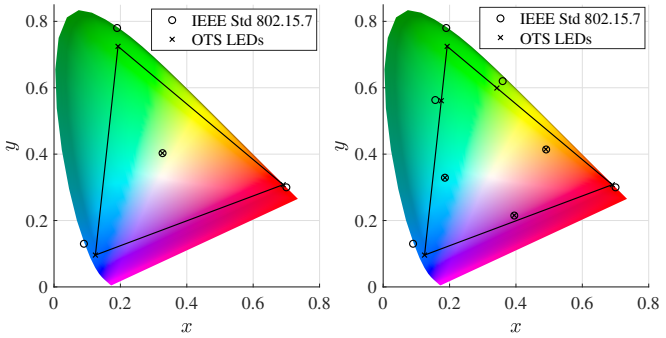


Fig. 2. CIE 1931 color matching functions (CMF) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ (left axis) and area-normalized spectral power distributions $S_{\mathrm{R}}(\lambda)$, $S_{\mathrm{G}}(\lambda)$, $S_{\mathrm{B}}(\lambda)$ of the colored OTS LEDs considered in [2] (right axis), all plotted versus wavelength λ [nm]. Both vertical axes are dimensionless: the color matching functions are the CIE 1931 standard-observer tristimulus weights, and the LED spectra are normalized so that each distribution integrates to unit area, *i.e.* $\int S_i\, d\lambda = 1$.

B. CSK Transmitter

In CSK, each symbol is represented by a chromaticity point in the CIE 1931 plane and is generated by appropriately combining the optical outputs of the colored LEDs [4]. IEEE Std 802.15.7 defines reference CSK constellations by selecting three light sources from seven predefined color bands, each identified by a 3-bit code [4]. Among the valid combinations, this work adopts the [100–010–001] basis because it provides the closest match to the chromaticities achievable with the OTS RGB LEDs considered here, consistent with the combinations recommended in [14].

Fig. 3 compares the IEEE reference symbols for 4-CSK and 8-CSK with the corresponding chromaticity points generated by the adopted OTS LEDs. The triangle delimits the color gamut achievable with the three adopted OTS LEDs. Because OTS LEDs exhibit broadband spectra rather than ideal monochromatic emissions, the constellation points realizable inside this gamut do not, in general, coincide exactly with the IEEE reference positions. Therefore, it indicates that practical optical generation introduces distortion with respect to the ideal IEEE reference constellation.



(a) 4-CSK (b) 8-CSK

Fig. 3. IEEE reference symbols and corresponding OTS-generated points for (a) 4-CSK and (b) 8-CSK in the CIE 1931 chromaticity diagram. Chromaticity coordinates x and y are dimensionless. The OTS points are obtained from measured LED spectra through Eqs. (1)–(2) and constrained nearest-symbol mapping. Distortion is already introduced at the optical signal generation stage.

Accordingly, the practical OTS constellation must be determined from the measured spectra of the red, green, and blue emitters. Let $S_{\mathrm{R}}(\lambda)$, $S_{\mathrm{G}}(\lambda)$, and $S_{\mathrm{B}}(\lambda)$ denote the measured spectral distributions of the three OTS LEDs reported in [2]. A candidate emitted spectrum is modeled as the linear combination:

$$S_{\mathbf{P}}(\lambda) = P_{\mathrm{R}} S_{\mathrm{R}}(\lambda) + P_{\mathrm{G}} S_{\mathrm{G}}(\lambda) + P_{\mathrm{B}} S_{\mathrm{B}}(\lambda), \tag{3}$$

where $\mathbf{P} = [P_{\mathrm{R}}\ P_{\mathrm{G}}\ P_{\mathrm{B}}]^{\mathrm{T}}$ is the vector of normalized electrical power coefficients associated with the red, green, and blue OTS LEDs, respectively. For each candidate $\mathbf{P} = [P_{\mathrm{R}}\ P_{\mathrm{G}}\ P_{\mathrm{B}}]^{\mathrm{T}}$, the resulting spectrum is converted into tristimulus values through Eq. (1) and then into chromaticity coordinates through Eq. (2).

For each IEEE reference symbol $\mathbf{s}_{\mathrm{ref}}$, the normalized RGB drive coefficients are obtained by solving

$$\mathbf{P}^{*} = \arg\min_{\mathbf{P}} \left\| \mathbf{s}_{\mathrm{ref}} - \mathbf{s}_{\mathbf{P}} \right\|_2 , \tag{4}$$

where $\mathbf{s}_{\mathbf{P}}$ is the chromaticity produced by the candidate OTS spectrum. The mapped point $\mathbf{s}_{\mathbf{P}^*}$ defines the corresponding practical OTS constellation point, whereas $\mathbf{P}^*$ provides the normalized RGB drive coefficients required to generate it.

This procedure explains why some IEEE reference points are not exactly realizable with OTS LEDs: finite spectral width and spectral overlap shift the achievable chromaticities away from their ideal target positions. Table I summarizes, for each symbol, the practical OTS chromaticity $[x\ y]$ and the associated normalized drive coefficients $(P_{\mathrm{R}}, P_{\mathrm{G}}, P_{\mathrm{B}})$. The left block corresponds to 4-CSK while the right block corresponds to 8-CSK. Bold chromaticity entries indicate practical OTS points that differ from the corresponding IEEE reference positions shown in Fig. 3.

TABLE I
POSITIONS OF THE SYMBOLS IN THE COLOR SPACE AND RESPECTIVE NORMALIZED ELECTRICAL POWERS OF THE OTS LEDS

4-CSK						8-CSK					
data	$[x\ y]$		P_{R}	P_{G}	P_{B}	data	$[x\ y]$		P_{R}	P_{G}	P_{B}
00	**0.193**	**0.725**	0	1	0	000	**0.193**	**0.725**	0	1	0
01	0.327	0.403	0.358	0.410	0.232	001	**0.175**	**0.561**	0	0.800	0.200
10	**0.096**	**0.125**	0	0	1	010	**0.343**	**0.599**	0.317	0.683	0
11	**0.690**	**0.308**	1	0	0	011	0.490	0.415	0.643	0.295	0.062
						100	**0.096**	**0.125**	0	0	1
						101	0.186	0.329	0.086	0.417	0.497
						110	0.395	0.216	0.572	0.034	0.394
						111	**0.690**	**0.308**	1	0	0

Bold $[x\ y]$ entries indicate OTS-feasible chromaticities that differ from the corresponding IEEE reference symbols.

After the normalized RGB power coefficients are determined, the corresponding forward currents of the OTS LEDs are obtained from the polynomial electrical power–current model reported in [15], namely,

$$\mathbf{I} = \begin{bmatrix} I_{\mathrm{R}} \\ I_{\mathrm{G}} \\ I_{\mathrm{B}} \end{bmatrix} = \begin{bmatrix} -6.174\times10^{-2} P_{\mathrm{R}}^2 + 4.855\times10^{-1} P_{\mathrm{R}} - 4.510\times10^{-3} \\ -3.471\times10^{-2} P_{\mathrm{G}}^2 + 3.586\times10^{-1} P_{\mathrm{G}} - 3.476\times10^{-3} \\ -3.271\times10^{-2} P_{\mathrm{B}}^2 + 3.471\times10^{-1} P_{\mathrm{B}} - 3.896\times10^{-3} \end{bmatrix}. \tag{5}$$

Therefore, the transmitter model is fully specified by the bit-to-symbol mapping in the chromaticity plane, the practical OTS constellation synthesis, and the conversion from normalized power coefficients to physical LED drive currents.

C. MIMO Channel

A common feature of VLC systems employing colored LEDs and OFs, such as the architecture in Fig. 1, is the lack of perfect orthogonality between color channels. Because the LED emission spectra are spread over wavelengths and the OFs are non-ideal, a given transmitted color component can leak into adjacent receiver branches, thereby producing inter-channel crosstalk [16], [17]. For this reason, the optical front-end is appropriately modeled as a multiple-input multiple-output (MIMO) channel.

The received voltage vector is thus written as

$$\mathbf{V} = \mathbf{H}\mathbf{I} + \mathbf{n}, \tag{6}$$

where $\mathbf{n} = [n_{\mathrm{R}}\ n_{\mathrm{G}}\ n_{\mathrm{B}}]^{\mathrm{T}}$ is a zero-mean additive white Gaussian noise (AWGN) vector with variance σ_{n}^2, and $\mathbf{H}$ is the channel matrix. The diagonal entries of $\mathbf{H}$ represent the direct gains

of the red, green, and blue branches, whereas the off-diagonal entries capture the corresponding crosstalk terms.

This work considers the same LEDs, OFs, and PDs as in [2], which, consequently, models the following channel matrix

$$\mathbf{H}' = \frac{1}{\Omega_{D_{1\mathrm{m}}}} \begin{bmatrix} 143.38 & 4.8802 & 0.7780 \\ 2.8135 & 153.53 & 53.289 \\ 1.7792 & 18.689 & 133.19 \end{bmatrix} \quad [\mathrm{mV/A}], \tag{7}$$

where $\Omega_{D_{1\mathrm{m}}} = 1.233 \times 10^{-5}$ denotes the gain corresponding to a receiver distance of $D = 1$ m, Lambertian order $n_{\mathrm{L}} = 10$, and PD area $A_{\mathrm{pd}} = 7.069 \times 10^{-2}$ cm^2, with $\phi = \theta = 0$, where ϕ represents the angle between the transmitter normal vector and the direction toward the receiver, and θ represents the angle between the receiver normal vector and the direction toward the transmitter [2]. In general, the channel matrix can be expressed as $\mathbf{H} = \Omega\mathbf{H}'$, where the gain is given by $\Omega = \frac{(n_{\mathrm{L}}+1)A_{\mathrm{pd}}}{2\pi D^2} \cos^{n_{\mathrm{L}}}(\phi) \cos(\theta)$.

D. CSK Receiver

At the receiver, the goal is to undo channel mixing, reconstruct the chromaticity of the received symbol, and then decide the transmitted class. IEEE Std 802.15.7 proposes the estimation of the CSI matrix $\hat{\mathbf{H}}$ followed by channel equalization (CE) in block III of Fig. 1. The resulting equalized RGB intensity vector is

$$\mathbf{V}_{\mathrm{CE}} = [R\,G\,B]^{\mathrm{T}} = \hat{\mathbf{H}}^{-1}\mathbf{V}. \tag{8}$$

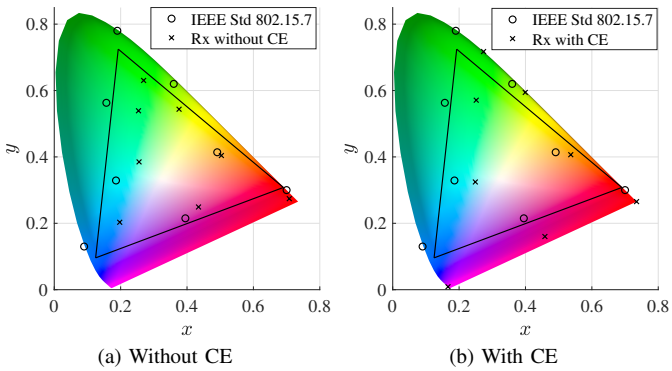
For block IV, although the standard specifies the transmitted CSK signal, it leaves the chromaticity reconstruction at the receiver side open, thus allowing some flexibility for exploring different approaches. In this work, the equalized RGB vector is explicitly mapped to the CIE 1931 tristimulus domain through

$$\mathbf{C}'_{\mathrm{CE}} = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{9}$$

after which Eq. (2) is applied to obtain the chromaticity estimate $\mathbf{s}'_{\mathrm{CE}} = [x'\,y']^{\mathrm{T}}$.

Fig. 4 shows the captured 8-CSK symbols without additive noise, both before Fig. 4(a) and after Fig. 4(b) CE. As expected, CE increases inter-symbol spacing and partially restores the intended constellation geometry. However, the equalized points still exhibit noticeable distortion when compared with the practical transmitter constellation shown in Fig. 3.

This residual distortion results from the combined effect of transmitter- and receiver-side spectral mismatches. As previously discussed, the practically generated constellation points do not exactly coincide with the ideal IEEE reference positions. In addition, the OFs used in the receiver do not exhibit transmittance curves $T_{\mathrm{R}}(\lambda)$, $T_{\mathrm{G}}(\lambda)$, and $T_{\mathrm{B}}(\lambda)$ identical to the CIE RGB color matching functions $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{b}(\lambda)$, as shown in Fig. 5. Therefore, the measured RGB components are obtained through practical optical responses at both the transmitter and receiver, rather than through ideal spectral responses. Consequently, even after channel equalization, the estimated RGB vector is only an approximation of the ideal CIE RGB tristimulus representation, and the reconstructed chromaticity $\mathbf{s}'_{\mathrm{CE}}$ remains distorted.



(a) Without CE (b) With CE

Fig. 4. 8-CSK symbols received without noise: (a) before CE and (b) after CE. Channel equalization increases inter-symbol spacing, but residual OF mismatch still introduces distortion relative to Fig. 3.

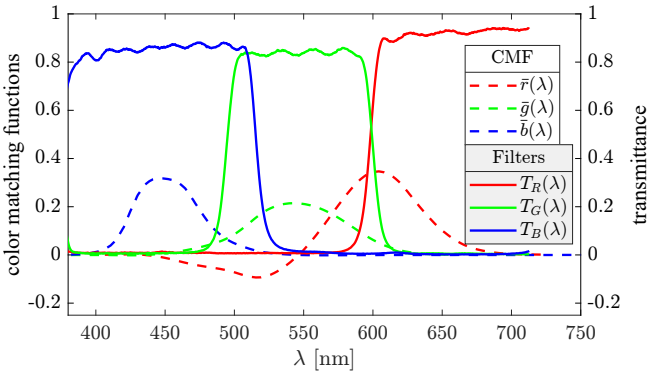


Fig. 5. CIE RGB color matching functions (CMF) $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ (left axis) and spectral transmittance $T_{\mathrm{R}}(\lambda)$, $T_{\mathrm{G}}(\lambda)$, $T_{\mathrm{B}}(\lambda)$ of the OFs considered in [2] (right axis), plotted versus wavelength λ [nm]. Both vertical axes are dimensionless: the color matching functions are spectral tristimulus weights, and the transmittance is the wavelength-resolved ratio of transmitted to incident optical power.

As the analytical benchmark for symbol detection in block V, this work adopts the MED rule

$$\hat{m}_{\mathrm{MED}} = \arg \min_{m \in \{1,\ldots,M\}} \left\| \mathbf{s}' - \mathbf{s}_m^{\mathrm{ref}} \right\|_2^2, \tag{10}$$

where $\mathbf{s}'$ is the received chromaticity vector and $\mathbf{s}_m^{\mathrm{ref}}$ is the nominal reference chromaticity of class m. Since the practical received symbols are distorted relative to the nominal constellation due to transmitter- and receiver-side spectral mismatch, the detector in (10) is not, in general, the maximum-likelihood detector for the actual received signal. Accordingly, it is adopted here as an ideal-reference analytical baseline for all symbol error rate (SER) comparisons.

III. Mitigating Distortions by Applying ML

Without compensation, decoding $\mathbf{s}'_{\mathrm{CE}}$ in block V is degraded by the receiver mismatch described in Section II. We therefore cast symbol detection as a supervised classification problem and evaluate whether ML methods can learn the resulting nonlinear decision regions [5]. Each model maps an input feature vector $\mathbf{f}$ to a class label $\hat{m}$, and all results are compared

against the analytical MED baseline defined in Section II (Eq. (10)). The 10 evaluated classifier families are briefly summarized below.

A. *Decision trees* (DT)

Hierarchical models used for classification or regression tasks that predict outcomes by performing a sequence of tests on input attributes at each internal node. Each branch represents the result of a test on an attribute, while each leaf node corresponds to a final prediction [18]. A decision tree begins with a single root node. The tree is then iteratively grown by evaluating potential splits of the leaf nodes. During each iteration, candidate splits are assessed using the loss metric. The split that yields the lowest loss is selected. If no candidate split provides a meaningful gain, the process halts for that node. This recursive procedure continues until no further beneficial splits are available or stopping criteria, such as maximum depth or minimum sample size, are met [19].

B. *Discriminant analysis* (DA)

This method derives linear combinations of the input variables that maximize the ratio of between-class to within-class variance (Fisher's criterion), thereby improving class separability [20], [21]. The resulting discriminant functions are then used to assign new observations to the most likely class.

C. *Logistic regression* (LR)

It models the mapping of a real value to the probability space $[0, 1]$ using a continuous, differentiable function called the logistic function [18], [19]. This function is defined as

$$\text{logistic}(\eta) = \frac{1}{1 + e^{-\eta}}, \tag{11}$$

where $\eta = \boldsymbol{\beta} \cdot \mathbf{f}$ is a linear combination of the input features $\mathbf{f}$ weighted by the parameters $\boldsymbol{\beta}$. The logistic regression training procedure consists of finding the optimal parameter values $\boldsymbol{\beta}$ that minimize the loss function [18], [19].

D. *Naive Bayes* (NB)

Naive Bayes classifies the input vector $\mathbf{f}$ by selecting the class that maximizes the posterior probability. This rule follows from Bayes' theorem under the naive assumption that features are conditionally independent given the class [18], [19]

$$\hat{m} = \underset{k \in \{1, \cdots, K\}}{\text{argmax}} \left(P(\Omega_k) \prod_{j=1}^{n_f} P(f_j \mid \Omega_k) \right), \tag{12}$$

where $P(\Omega_k)$ is the prior probability of class Ω_k, $P(f_j \mid \Omega_k)$ is the likelihood of the j-th feature given class Ω_k, and the product follows from the (naive) conditional-independence assumption.

E. *Support vector machines* (SVM)

Supervised learning algorithms are designed to identify an optimal hyperplane that separates data points from different classes with the maximum possible margin in a high-dimensional feature space [18], [19]. Formally, SVM constructs this hyperplane to maximize the distance to the nearest training samples of any class, known as support vectors, thus enhancing the generalization capability and robustness of the model [18].

F. *k-nearest neighbors* (kNN)

kNN is a non-parametric, instance-based learning method. For a given input vector $\mathbf{f}$, it identifies the k closest samples in the training set, typically under Euclidean distance, and predicts the class by majority vote among those neighbors [18].

G. *Ensemble models* (EM)

Combine the predictions of multiple individual models to produce a more robust and accurate final prediction. The core concept is to aggregate the outputs through voting to reduce the likelihood of misclassification that may occur when relying on a single model [18]. One of the most used ensemble methods is boosting. It operates by sequentially training simple models, where each subsequent model attempts to correct the errors of its predecessor. This iterative process assigns higher weights to data points misclassified by previous models, thereby forcing the next learner to focus on these harder cases [22], [18].

H. *Feedforward neural networks* (FNNs)

Consist of processing units (neurons) organized into a directed acyclic graph of layers. Each neuron j computes an affine transformation of its inputs, followed by a non-linear activation function. For a given layer l, the activation of the j-th unit, denoted as $a_j^{(l)}$, is computed as [21], [18]

$$a_j^{(l)} = g\left(\sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)} \right), \tag{13}$$

where $w_{jk}^{(l)}$ and $b_j^{(l)}$ represent the weights and bias associated with the l-th layer, and $g(\cdot)$ denotes a non-linear activation function (e.g., ReLU, sigmoid). Model training is formulated as an empirical risk minimization (ERM) problem, wherein the optimal parameters $\theta = \{W, b\}$ are estimated by minimizing a cost function $\mathcal{L}$ over the training dataset [21], [18].

I. *Kernel approximation* (KA)

Since linear separators often underperform on non-linearly separable data, KA methods utilize a feature mapping function $\psi : \mathcal{X} \to \mathcal{F}$ to embed input features from the original domain $\mathcal{X}$ into a higher-dimensional feature space $\mathcal{F}$ [18], [19]. Given a labeled dataset S, the objective is to train a linear predictor within this transformed space. The efficacy of this paradigm relies on selecting a mapping ψ that renders the data distribution linearly separable in $\mathcal{F}$, thereby optimizing the learner's performance [19]. The most well-known examples of kernel methods are the support vector machines.

J. Kolmogorov-Arnold Networks (KAN)

Provide an alternative architecture to classical FNNs [23]. Unlike traditional models that rely on fixed activation functions at each neuron and learnable linear weights, KANs parameterize the connections between neurons with learnable activation functions (typically B-splines). This design allows KANs to achieve high approximation accuracy with significantly fewer parameters than standard FNNs [23]. Formally, a KAN consists of L layers. Let $\mathbf{h}^{(l)} \in \mathbb{R}^{n_l}$ denote the state vector at layer l (to avoid conflict with chromaticity coordinate x). The transition to the subsequent layer $l+1$ is computed via a matrix of learnable univariate functions $\mathbf{\Phi}_l$ [23]

$$h_i^{(l+1)} = \sum_{j=1}^{n_l} \phi_{l,j,i}\left(h_j^{(l)}\right), \quad i = 1, \cdots, n_{l+1}, \tag{14}$$

where $\phi_{l,j,i} : \mathbb{R} \to \mathbb{R}$ is the activation function connecting the j-th neuron of layer l to the i-th neuron of layer $l+1$.

In this work, we employ the ReLU-KAN variant [24], which replaces spline evaluations with trainable basis functions that are more efficient for matrix-based computation. This choice reduces the computational overhead of standard KAN implementations while preserving the adaptive edge-wise nonlinearities that motivates their use.

IV. Numerical Results

This section evaluates 10 ML classifier families under multiple hyperparameter settings, yielding 37 trained models for CSK symbol recovery over the non-ideal VLC channel described in Section II. The input feature vectors $\mathbf{f}$ are extracted from six signal representations, defined in Table II: $\mathbf{V}_{\text{CE}}$, $\mathbf{C}'_{\text{CE}}$, and $\mathbf{s}'_{\text{CE}}$ with CE; and $\mathbf{V}$, $\mathbf{C}'$, and $\mathbf{s}'$ without CE, where $\mathbf{V}$ denotes the raw voltage vector before channel equalization.

The noise budget is established under the following baseline conditions: $\phi = \theta = 0$, $D = 1$ m, and a TIA gain of $G_{\text{TIA}} = 10$ kV/A [16]. Under these conditions, the dominant noise components and their variances are: thermal noise ($\sigma^2_{\text{TH}} = 4.60 \times 10^{-12}$ V^2), shot noise due to dark current ($\sigma^2_{\text{DC}} = 1.60 \times 10^{-18}$ V^2), shot noise induced by the received communication signal ($\sigma^2_{\text{RS}} = 5.73 \times 10^{-13}$ V^2), and shot noise due to background irradiance[1] ($\sigma^2_{\text{BG}} = 1.75 \times 10^{-10}$ V^2) [16]. Background radiation is thus the dominant noise source [6], with $\sigma^2_{\text{BG,dB}} = -102.01$ dBV^2, and the aggregate noise variance is $\sigma^2_{\text{n,dB}} = -101.67$ dBV^2.

Figure 6 reports the average SER over 10 realizations for these feature sets in 4-CSK and 8-CSK at $\phi = 76.02°$ and $\phi = 73.11°$, respectively[2].

These operating points correspond to intermediate channel conditions in which neither noise nor signal distortion effects are negligible. This regime avoids both saturation (near-zero SER) and fully degraded performance, thereby enabling a more discriminative comparison across models and feature representations under comparable channel gains.

[1]Computed assuming a visible-range solar irradiance of 534.6 W/m² [25].

[2]Conditions with equivalent channel gains also arise from other combinations of D, ϕ, and θ.

TABLE II
Notation for features used in ML models

Symb.	Definition	Dim.	Notes
$\mathbf{V}_{\text{CE}}$	RGB intensity vector	3D	Channel equalized
$\mathbf{V}$	RGB intensity vector	3D	Raw (without CE)
$\mathbf{C}'_{\text{CE}}$	CIE 1931 tristimulus	3D	Computed from $\mathbf{V}_{\text{CE}}$
$\mathbf{C}'$	CIE 1931 tristimulus	3D	Computed from $\mathbf{V}$
$\mathbf{s}'_{\text{CE}}$	CIE 1931 chromaticity coordinates	2D	Projected from $\mathbf{C}'_{\text{CE}}$
$\mathbf{s}'$	CIE 1931 chromaticity coordinates	2D	Projected from $\mathbf{C}'$

A. SER Performance Comparison

Fig. 6 provides a global comparison of the average SER obtained by all 37 evaluated ML models over the six feature representations, for both 4-CSK and 8-CSK. Based on these results, Tables III and IV summarize the best-performing configuration within each classifier family, ranked by ascending order of SER. The complete hyperparameter grid corresponding to all evaluated configurations is reported in Appendix A (Tables V and VI).

For 4-CSK (Table III), KAN-3 with feature $\mathbf{V}$ achieves the minimum SER of 10.78%. The second-best family is LR at 12.03%, meaning KAN-3 provides a 10.4% relative reduction in SER. Compared to the best FNN result (19.84%), KAN achieves a relative reduction of 45.7%. For 8-CSK (Table IV), KAN-2 with feature $\mathbf{V}$ achieves an SER of 9.06%, followed by DA-1 at 10.78% and EM-3 at 11.56%. Relative to DA-1, KAN-2 provides a 16.0% reduction in SER. Compared with the best FNN result, KAN achieves a relative reduction of 53.3%.

The results suggest that the observed performance gains arise from two complementary effects. First, several relatively simple learning models, including LR, DA, NB, and EM, already achieve strong performance. This indicates that a substantial portion of the error originates from the mismatch between the nominal reference constellation and the actual received class distribution, rather than from limited classifier complexity alone. Second, the additional improvement provided by KAN suggests that the residual class structure in the selected feature space remains nonlinear, even after this mismatch is implicitly compensated through data-driven learning.

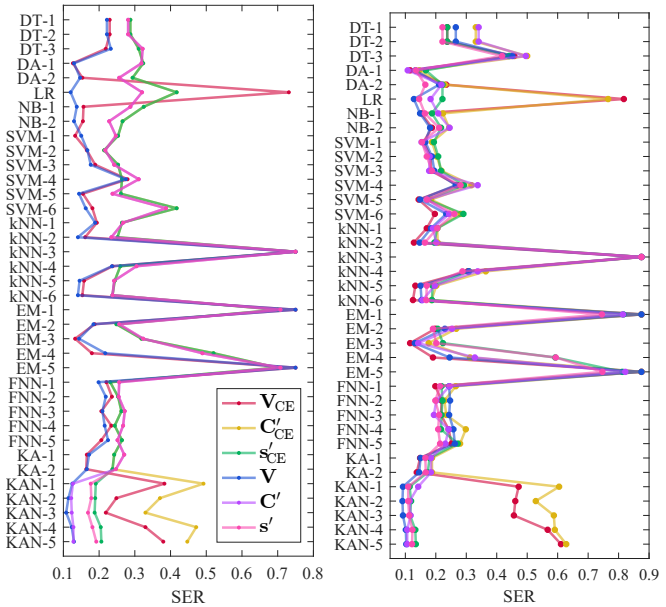
Besides that, a consistent performance gap is observed between KAN and FNN configurations in this setup. In the best per-family results, FNN achieves 19.84% (4-CSK) and 19.38% (8-CSK), whereas KAN reaches 10.78% and 9.06%, respectively. These values confirm that KAN yields lower SER under identical data-generation and evaluation conditions. The larger KAN-to-FNN reduction in 8-CSK than in 4-CSK (53.3% versus 45.7%) further suggests that the advantage of a more flexible nonlinear models becomes more pronounced as the decision structure becomes more intricate.

The full hyperparameter grids in Appendix A provide an additional structural insight. For both modulation orders, the best KAN configurations are shallow (KAN-3 for 4-CSK and KAN-2 for 8-CSK), whereas adding extra layers does not improve the best observed SER. This suggests that, for the distortion patterns considered here, performance depends more on learning an appropriate nonlinear feature transformation

than on increasing network depth. In practical terms, the best-performing KANs are not only more accurate than the tested FNNs, but also structurally compact.

Regarding feature selection, the results indicate that $\mathbf{V}$-based features are selected most frequently among the best family entries (8/10 families in 4-CSK and 5/10 in 8-CSK), with $\mathbf{V}_{\mathrm{CE}}$ selected in three specific cases (DA and DT in 4-CSK and kNN in 8-CSK). Projected chromaticity/tristimulus features remain competitive for some linear models (e.g., DA-1 with $\mathbf{C}'$ in 8-CSK), while the minimum-SER configurations in both modulations use 3D intensity features. This pattern indicates that channel equalization and 2D projection are not uniformly beneficial for classification: when mismatch remains, the raw 3D intensity vector often preserves discriminative information that is partially suppressed by idealized receiver transformations.

With respect to computational cost, the results show a clear distinction between training and inference time. The proposed KAN models required 1563.6–1897.8 ms for training in their best configurations, which is lower than several competing families, such as DA (2468.8–10595.2 ms), SVM (7985.9–10163.1 ms), EM (6596.9–8320.4 ms), and KA (6092.2–8098.5 ms). During inference, KAN required 0.8564 ms for 8-CSK and 1.7444 ms for 4-CSK, which is higher than DA, LR, DT, and FNN, but still below EM. In particular, DT and FNN achieved the lowest inference times, between 0.1225 and 0.1472 ms, while DA and LR remained below 0.5 ms. By contrast, EM exhibits the highest inference cost, reaching 2.4557 ms for 8-CSK and 2.5743 ms for 4-CSK. Therefore, the timing analysis confirms that KAN provides moderate training cost and acceptable inference time, while simpler models remain preferable only when minimum latency is the dominant requirement.



(a) 4-CSK for $\phi = 76.02°$ (b) 8-CSK for $\phi = 73.11°$

Fig. 6. Average SER (dimensionless, $0 \leq \mathrm{SER} \leq 1$) obtained by the 37 evaluated ML models over the six considered feature representations for (a) 4-CSK and (b) 8-CSK.

TABLE III
BEST-PERFORMING MODEL PER CLASSIFIER FAMILY FOR 4-CSK, RANKED BY LOWEST SER.

Family	Best model	SER	Best feature	Train. [ms]	Infer. [ms]
KAN	KAN-3	0.1078	$\mathbf{V}$	1897.8	1.7444
LR	LR	0.1203	$\mathbf{V}$	3897.8	0.1937
DA	DA-1	0.1266	$\mathbf{V}_{\mathrm{CE}}$	10595.2	0.2362
NB	NB-2	0.1297	$\mathbf{V}$	3804.8	0.4102
EM	EM-3	0.1328	$\mathbf{V}_{\mathrm{CE}}$	8320.4	2.5743
SVM	SVM-1	0.1328	$\mathbf{V}_{\mathrm{CE}}$	7985.9	0.3033
kNN	kNN-2*	0.1406	$\mathbf{V}$	4150.0	0.3079
KA	KA-2	0.1641	$\mathbf{V}_{\mathrm{CE}}$	6092.2	0.4613
FNN	FNN-1	0.1984	$\mathbf{V}$	4172.6	0.1358
DT	DT-3	0.2188	$\mathbf{V}_{\mathrm{CE}}$	5785.1	0.1353

*kNN-2 and kNN-6 are tied for 4-CSK (SER $= 0.1406$).

TABLE IV
BEST-PERFORMING MODEL PER CLASSIFIER FAMILY FOR 8-CSK, RANKED BY LOWEST SER.

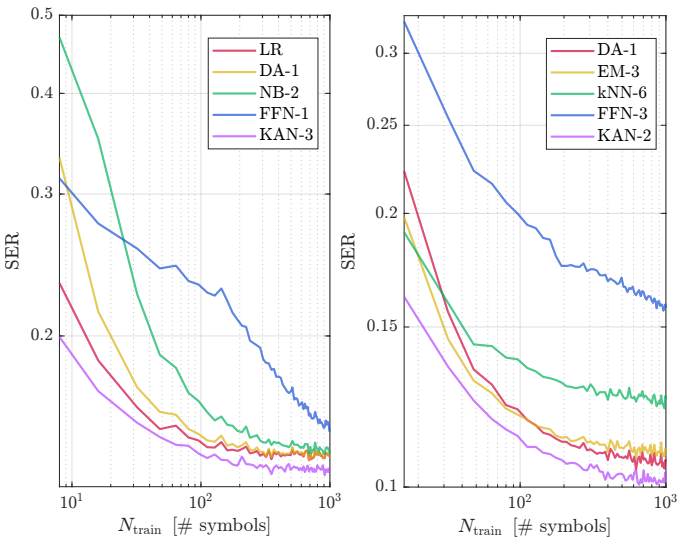
Family	Best model	SER	Best feature	Train. [ms]	Infer. [ms]
KAN	KAN-2	0.0906	$\mathbf{V}$	1563.6	0.8564
DA	DA-1	0.1078	$\mathbf{C}'$	2468.8	0.1698
EM	EM-3	0.1156	$\mathbf{V}_{\mathrm{CE}}$	6596.9	2.4557
kNN	kNN-6	0.1250	$\mathbf{V}_{\mathrm{CE}}$	3434.2	0.4705
LR	LR	0.1281	$\mathbf{V}$	2665.7	0.4523
KA	KA-2	0.1375	$\mathbf{V}_{\mathrm{CE}}$	8098.5	1.4941
SVM	SVM-5	0.1438	$\mathbf{V}_{\mathrm{CE}}$	10163.1	1.0267
NB	NB-1	0.1484	$\mathbf{V}$	2632.4	0.2720
FNN	FNN-3	0.1938	$\mathbf{C}'$	2839.7	0.1472
DT	DT-1	0.2219	$\mathbf{s}'$	5440.6	0.1225

The impact of the *size of the training set* on performance, illustrated in Fig. 7, reveals fundamental differences in the learning dynamics of the evaluated architectures.

In the data-scarce regime (e.g., $N_{\mathrm{train}} \leq 16$ symbols), simple linear classifiers, such as LR and DA, demonstrate high sample efficiency, rapidly establishing reasonable decision boundaries. In contrast, standard FNNs struggle significantly in this low-data setting, yielding error rates nearly twice as high as those of the strongest classical baselines (e.g., exceeding 32% for 8-CSK). This behavior confirms the well-known limitation of conventional deep learning architectures, which typically require large datasets to overcome initialization variance and avoid overfitting.

As the training set increases, distinct saturation behaviors emerge. Linear models quickly reach an asymptotic error floor (approximately 12% for 8-CSK), indicating that their performance is bounded by model bias, specifically, their inability to capture the non-linear curvature of the optimal decision boundaries. However, the KAN architecture exhibits a unique dual advantage. It mirrors the rapid convergence of linear models in the early learning phase, (*e.g.*, approximately 16.3% at $N = 16$), while retaining the capacity to further reduce error as more data become available, eventually surpassing the linear floor to reach approximately 10% with full training set. In this sense, KAN combines better sample efficiency than the evaluated FNNs with sufficient nonlinear flexibility to reduce the asymptotic error floor in this problem.

Fig. 8a shows the SER as a function of the noise variance σ_{n}^2 for 4-CSK, whereas Fig. 8b presents the corresponding results for 8-CSK. In both cases, σ_{n}^2 is expressed in dBV^2, i.e. $\sigma_{\mathrm{n,dB}}^2 = 10\log_{10}\frac{\sigma_{\mathrm{n}}^2}{1\mathrm{V}^2}$. For a fixed received signal power $P_{\mathrm{sig,dB}}$, also expressed in dBV^2, the corresponding electrical signal-



(a) 4-CSK, $\phi = 76.02°$ (b) 8-CSK, $\phi = 73.11°$

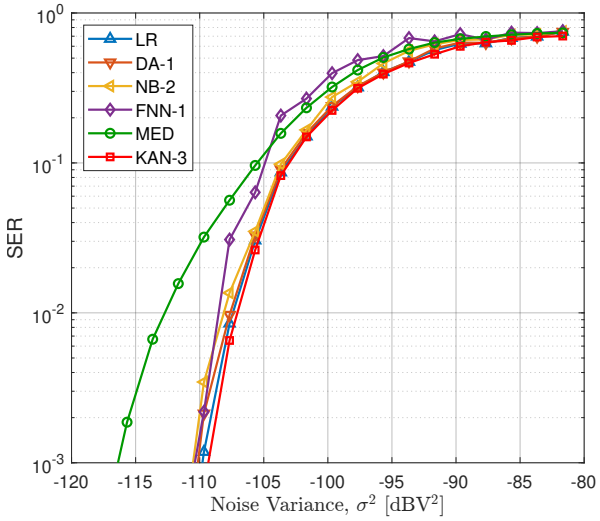
Fig. 7. Average SER (dimensionless, $0 \leq \text{SER} \leq 1$) as a function of the number of pilot training symbols N for (a) 4-CSK at $\phi = 76.02°$ and (b) 8-CSK at $\phi = 73.11°$. Both axes are dimensionless: SER is the ratio of erroneously detected symbols to the total transmitted, and N is a discrete count. Each curve represents a distinct ML classifier family: KAN, FNN, LR, NB, DA, and the MED baseline. Results are averaged over 100 independent realizations.

to-noise ratio (SNR) is given by $\text{SNR}_{\text{dB}} = P_{\text{sig,dB}} - \sigma^2_{\text{n,dB}}$. Hence, decreasing $\sigma^2_{\text{n,dB}}$ corresponds to increasing SNR, and the two figures illustrate how the detectors behave over the considered operating range for each modulation order.

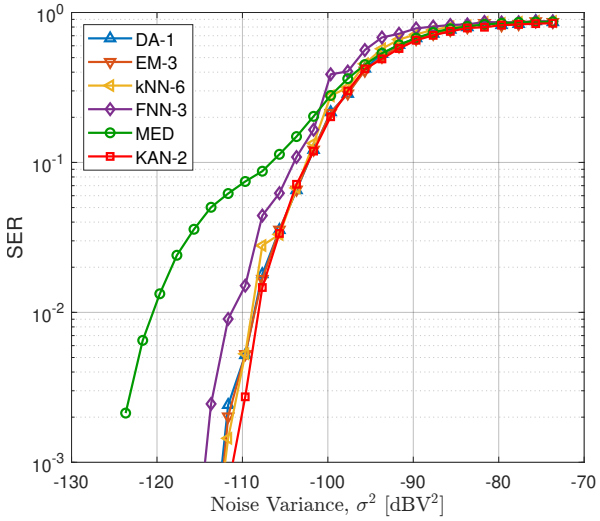
For 4-CSK (Fig. 8a) when $\sigma^2_{\text{n,dB}} < -103.67$ dBV^2, all learned classifiers outperform the MED baseline of Eq. (10). At $\sigma^2_{\text{n,dB}} = -101.67$ dBV^2, KAN-3 achieves a SER of 14.9%, compared with 16.4% for the best linear baseline and 23.3% for MED. At $\sigma^2_{\text{n,dB}} = -105.67$ dBV^2, KAN-3 further decreases to 2.6%, while MED remains above 10%. In terms of a fixed SER target, the KAN curve is shifted by approximately 4 dB relative to MED within the tested range, indicating better robustness to the distorted decision geometry.

For 8-CSK (Fig. 8b), the same trend holds, although the performance gap is smaller in the highest-noise regime. For $\sigma^2_{\text{n,dB}} \geq -97.67$ dBV^2, KAN and the best linear discriminator are nearly indistinguishable (29.9% versus 30.0%), which suggests that additive noise dominates performance in this regime. As the noise variance decreases, model mismatch becomes more pronounced. At $\sigma^2_{\text{n}} = -105.67$ dBV^2, KAN-2 achieves a SER of 3.36%, whereas the best linear classifier (DA) remains at 4.77%, corresponding to a relative reduction of approximately 30%.

Taken together, these results highlight two distinct operating regimes. In the noise-limited regime, detector performance converges and architectural differences matter little. In the lower-noise regime, however, residual spectral and receiver mismatch becomes the dominant limiting factor, and the ranking is then determined by how effectively each model captures the resulting distorted decision boundaries.



(a) 4-CSK: best KAN-3 versus the MED baseline and the top-performing classifiers (FNN-1, LR, NB-2, DA-1).



(b) 8-CSK: best KAN-2 versus the MED baseline and the top-performing classifiers (FNN-3, DA-1, kNN-6, EM-3).

Fig. 8. SER as a function of noise variance σ^2_{n} [dBV^2] for (a) 4-CSK and (b) 8-CSK. Each panel compares the best KAN against the MED baseline and the top-performing classifiers from other families.

V. Discussion of Results

A. Key Explanations for the Observed Gains

The MED detector of Eq. (10) assumes the IEEE reference constellation geometry and therefore constitutes a mismatched baseline in the presence of OTS-induced spectral spreading, filter/PD non-idealities, and nonlinear electro-optical conversion. These effects shift and warp class clusters relative to the reference points, whereas learned classifiers adapt their decision regions directly from the distorted data and consequently achieve lower SER, as reported in Section IV.

B. KAN versus FNN

KAN models employ learnable edge-wise nonlinear functions, whereas FNNs rely on fixed activations with learned weights (cf. Section III). For the distortion patterns observed

in this CSK channel, the added functional flexibility of KAN appears better aligned with the data, which is consistent with the lower SER values reported in Section IV.

C. Feature-Space Discussion and Practical Implication

Results also confirm that retaining 3D information is critical. Features based on $\mathbf{V}$ (and, depending on the model, $\mathbf{C}'$) preserve more discriminative structure than purely projected chromaticity coordinates, since luminance-related information is not discarded. This explains the recurring dominance of $\mathbf{V}$-based models in the ranking tables and the superior performance achieved by KAN.

From an implementation perspective, the SER gains of KAN must be weighed against its higher computational and memory cost, due to the learnable basis-function parameters. For deployment, the most relevant quantitative metrics are inference time per symbol and parameter count, since a real-time system must be able to detect symbols promptly. However, training time should also be taken into account when the receiver is expected to self-adjust to a given physical configuration of LEDs, OFs, and PDs. In that case, retraining is no longer purely an offline design step, but part of the adaptation procedure itself, so excessively large training cost may limit practical adaptability.

VI. Conclusion and Future Work

This paper studied CSK symbol detection in a realistic OTS-VLC receiver, where LED spectral broadening and receiver spectral mismatch distort the constellation relative to the IEEE reference geometry. The system model adopts the $[100-010-001]$ basis convention [14], defines an explicit receiver mapping, and details how the OTS constellation is generated from measured spectra. Under this framework, 10 classifier families (37 configurations) were evaluated against the MED benchmark.

Across both modulation formats, KAN consistently achieved the best performance among the evaluated classifier families under the considered conditions. Relative to the best FNN configurations, these results correspond to SER reductions of 45.7% and 53.3%, respectively, with the larger gain in 8-CSK suggesting that KAN becomes more advantageous as the decision geometry becomes more intricate. The results also support three broader conclusions. First, a significant portion of the performance loss of the analytical MED detector arises from reference mismatch, as even simple learned families substantially outperform it. Second, the additional gains achived by KAN indicate that the underlying class boundaries remain nonlinear after mismatch compensation. Third, the repeated preference for $\mathbf{V}$-based features indicates that preserving the full 3D intensity information is usually more effective than projecting the received signal prematurely into 2D chromaticity coordinates.

A further practical implication is that the best-performing KANs in this study are shallow rather than deep, which indicates that the relevant compensation mechanism is a compact nonlinear remapping of the distorted feature space, rather than a highly layered architecture. This is encouraging from an implementation perspective, as it indicates that the observed gains do not inherently depend on excessively large models. These findings are limited to the conditions studied here, namely simulated links parameterized by measured OTS components, 4-CSK/8-CSK constellations, and the tested noise and channel ranges. Within this scope, however, they support a practical receiver-design insight: in OTS-CSK links, performance is limited not only by channel equalization, but also by how well the detector captures the distorted decision geometry while retaining informative 3D measurements. The most relevant next steps are therefore experimental validation with hardware, explicit deployment metrics such as inference time and parameter count, and evaluation under higher-order constellations and time-varying channels. These extensions will determine whether the SER gains observed here can be translated into robust real-time implementations.

Appendix A
Complete Hyperparameter Grid

For transparency and reproducibility, the complete hyperparameter sweep (all 37 trained models) is reported in Tables V and VI.

TABLE V
Complete hyperparameter grid and outcomes for 4-CSK.

Model	Hyperparameters	SER	σ^2_{SER} $[\times 10^{-4}]$	Feature	Train. time [ms]	Infer. time [ms]
DT-1	Splits = 100	0.2219	47.6	$\mathbf{V}$	5491.7	0.1121
DT-2	Splits = 20	0.2219	47.6	$\mathbf{V}$	4429.9	0.1400
DT-3	Splits = 4	0.2188	21.7	$\mathbf{V}_{\mathrm{CE}}$	5785.1	0.1353
DA-1	Linear discriminant	0.1266	19.8	$\mathbf{V}_{\mathrm{CE}}$	10595.2	0.2362
DA-2	Quadratic discriminant	0.1469	8.8	$\mathbf{V}$	3740.9	0.2236
LR	Coeff. tol. $= 10^{-4}$	**0.1203**	4.9	$\mathbf{V}$	3897.8	0.1937
NB-1	Gaussian pred.	0.1375	6.5	$\mathbf{V}$	3336.8	0.1838
NB-2	Kernel-unb. Gaussian pred.	0.1297	21.1	$\mathbf{V}$	3804.8	0.4102
SVM-1	Linear, auto scale	0.1328	24.1	$\mathbf{V}_{\mathrm{CE}}$	7985.9	0.3033
SVM-2	Quadratic, auto scale	0.1656	18.0	$\mathbf{V}_{\mathrm{CE}}$	7596.1	0.3253
SVM-3	Cubic, auto scale	0.1766	12.0	$\mathbf{V}$	3984.7	0.2831
SVM-4	Gaussian, $\gamma=0.35$	0.2406	67.3	$\mathbf{s}'_{\mathrm{CE}}$	3121.9	0.2322
SVM-5	Gaussian, $\gamma=1.4$	0.1438	35.9	$\mathbf{V}$	4253.5	0.2515
SVM-6	Gaussian, $\gamma=3.7$	0.1625	17.8	$\mathbf{V}$	3679.8	0.2354
kNN-1	$k=1$, Euclidean, equal weights	0.1891	16.9	$\mathbf{V}$	3608.5	0.2902
kNN-2	$k=10$, Euclidean, equal weights	0.1406	18.7	$\mathbf{V}$	4150.0	0.3079
kNN-3	$k=100$, Euclidean, equal weights	0.7500	24.4	$\mathbf{V}_{\mathrm{CE}}$	7276.7	0.3272
kNN-4	$k=10$, cosine, equal weights	0.2359	0.0	$\mathbf{V}$	3417.1	0.3265
kNN-5	$k=10$, Minkowski, equal weights	0.1453	12.2	$\mathbf{V}$	3366.5	0.3590
kNN-6	$k=10$, Euclidean, squared inverse weights	0.1406	14.1	$\mathbf{V}$	3993.8	0.3097
EM-1	AdaBoost/DT (20 splits)	0.7000	48.5	$\mathbf{s}'_{\mathrm{CE}}$	3405.0	0.1166
EM-2	Bagging/DT (63 splits)	0.1844	0.0	$\mathbf{V}$	4086.3	1.0674
EM-3	Subspace/DA	0.1328	30.4	$\mathbf{V}_{\mathrm{CE}}$	8320.4	2.5743
EM-4	Subspace/kNN	0.1797	18.6	$\mathbf{V}_{\mathrm{CE}}$	8164.7	4.6161
EM-5	RUSBoost/DT (20 splits)	0.6984	76.5	$\mathbf{s}'_{\mathrm{CE}}$	4227.6	0.1202
FNN-1	1 layer, 10 neurons	0.1984	0.0	$\mathbf{V}$	4172.6	0.1358
FNN-2	1 layer, 25 neurons	0.2188	41.8	$\mathbf{V}$	3997.1	0.1134
FNN-3	1 layer, 100 neurons	0.2062	49.6	$\mathbf{V}_{\mathrm{CE}}$	4740.6	0.1678
FNN-4	2 layers, [10 10] neurons	0.2156	35.4	$\mathbf{V}$	3645.1	0.1128
FNN-5	3 layers, [10 10 10] neurons	0.2062	48.8	$\mathbf{V}_{\mathrm{CE}}$	5259.2	0.1799
KA-1	SVM, 28 learn., hinge	0.1656	13.5	$\mathbf{V}_{\mathrm{CE}}$	6309.3	0.5581
KA-2	LR, 28 learn., quadratic	0.1641	28.3	$\mathbf{V}_{\mathrm{CE}}$	6092.2	0.4613
KAN-1	1 layer, 10 nodes	**0.1250**	15.62	$\mathbf{V}$	1570.5	0.7220
KAN-2	1 layer, 25 nodes	**0.1141**	14.18	$\mathbf{V}$	1627.1	0.9210
KAN-3	1 layer, 100 nodes	**0.1078**	5.10	$\mathbf{V}$	1897.8	1.7444
KAN-4	2 layers, [10 10] nodes	**0.1234**	17.80	$\mathbf{V}$	2207.4	1.0514
KAN-5	3 layers, [10 10 10] nodes	0.1281	15.53	$\mathbf{C}'$	2866.6	1.5500

The 5 smallest SER values are in bold. All ensemble methods use 30 learning cycles. All FNN models use ReLU activations. All KAN models use learnable ReLU, basis order 4, and grid size 3.

Acknowledgment

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) of Brazil

TABLE VI
COMPLETE HYPERPARAMETER GRID AND OUTCOMES FOR 8-CSK.

Model	Hyperparameters	SER	σ^2_{SER} [$\times 10^{-4}$]	Feature	Train. time [ms]	Infer. time [ms]
DT-1	Splits = 100	0.2219	41.7	$\mathbf{s}'$	5440.6	0.1225
DT-2	Splits = 20	0.2219	41.7	$\mathbf{s}'$	1795.6	0.1259
DT-3	Splits = 4	0.4172	9.8	$\mathbf{s}'$	2774.9	0.1510
DA-1	Linear discriminant	**0.1078**	14.9	$\mathbf{C}'$	2468.8	0.1698
DA-2	Quadratic discriminant	0.1656	43.5	$\mathbf{s}'$	1991.9	0.2709
LR	Coeff. tol. $= 10^{-4}$	0.1281	24.8	$\mathbf{V}$	2665.7	0.4523
NB-1	Gaussian pred.	0.1484	9.2	$\mathbf{V}$	2632.4	0.2720
NB-2	Kernel-unb. Gaussian pred.	0.1812	36.5	$\mathbf{V}$	3299.8	0.8191
SVM-1	Linear, auto scale	0.1531	57.5	$\mathbf{C}'$	3062.8	0.5245
SVM-2	Quadratic, auto scale	0.1703	45.3	$\mathbf{s}'$	3849.1	0.7094
SVM-3	Cubic, auto scale	0.1781	27.3	$\mathbf{C}'$	3098.2	0.5736
SVM-4	Gaussian, $\gamma = 0.35$	0.2656	34.9	$\mathbf{V}$	3840.1	0.6615
SVM-5	Gaussian, $\gamma = 1.4$	0.1438	29.4	$\mathbf{V}_{\text{CE}}$	10163.1	1.0267
SVM-6	Gaussian, $\gamma = 3.7$	0.1969	26.5	$\mathbf{V}_{\text{CE}}$	10039.9	0.9058
kNN-1	$k = 1$, Euclidean, equal weights	0.1703	27.8	$\mathbf{V}_{\text{CE}}$	7837.7	0.6848
kNN-2	$k = 10$, Euclidean, equal weights	0.1281	31.7	$\mathbf{V}_{\text{CE}}$	7333.7	0.5076
kNN-3	$k = 100$, Euclidean, equal weights	0.8750	13.5	$\mathbf{V}_{\text{CE}}$	6918.3	0.4894
kNN-4	$k = 10$, cosine, equal weights	0.2875	0.0	$\mathbf{s}'$	5387.8	0.3851
kNN-5	$k = 10$, Minkowski, equal weights	0.1328	34.3	$\mathbf{V}_{\text{CE}}$	7146.0	0.3994
kNN-6	$k = 10$, Euclidean, squared inverse weights	0.1250	14.2	$\mathbf{V}_{\text{CE}}$	3434.2	0.4705
EM-1	AdaBoost/DT (20 splits)	0.7453	20.5	$\mathbf{s}'$	2920.1	0.1470
EM-2	Bagging/DT (63 splits)	0.1906	747.6	$\mathbf{s}'$	3890.5	1.6329
EM-3	Subspace/DA	0.1156	24.4	$\mathbf{V}_{\text{CE}}$	6596.9	2.4557
EM-4	Subspace/kNN	0.1906	16.4	$\mathbf{V}_{\text{CE}}$	8504.6	4.1915
EM-5	RUSBoost/DT (20 splits)	0.7469	34.2	$\mathbf{s}'$	3776.1	0.1463
FNN-1	1 layer, 10 neurons	0.1984	0.0	$\mathbf{V}_{\text{CE}}$	5831.8	0.3063
FNN-2	1 layer, 25 neurons	0.1984	32.0	$\mathbf{s}'$	4456.2	0.1882
FNN-3	1 layer, 100 neurons	0.1938	24.7	$\mathbf{C}'$	2839.7	0.1472
FNN-4	2 layers, [10 10] neurons	0.2078	36.5	$\mathbf{s}'$	3812.0	0.1422
FNN-5	3 layers, [10 10 10] neurons	0.2141	18.5	$\mathbf{s}'$	5567.0	0.1759
KA-1	SVM, 28 learn., hinge	0.1484	49.6	$\mathbf{V}_{\text{CE}}$	8868.0	1.4221
KA-2	LR, 28 learn., quadratic	0.1375	20.2	$\mathbf{V}_{\text{CE}}$	8098.5	1.4941
KAN-1	1 layer, 10 nodes	**0.0922**	6.57	$\mathbf{V}$	1548.9	0.7935
KAN-2	1 layer, 25 nodes	**0.0906**	3.81	$\mathbf{V}$	1563.6	0.8564
KAN-3	1 layer, 100 nodes	**0.0922**	7.54	$\mathbf{V}$	1931.9	1.6882
KAN-4	2 layers, [10 10] nodes	**0.1016**	5.98	$\mathbf{V}$	2237.2	1.1065
KAN-5	3 layers, [10 10 10] nodes	0.1031	17.68	$\mathbf{V}$	2885.1	1.5636

The 5 smallest SER values are in bold. All ensemble methods use 30 learning cycles. All FNN models use ReLU activations. All KAN models use learnable ReLU, basis order 4, and grid size 3.

under Grant 314618/2023-6 and by the Coordination for the Improvement of Higher Education Personnel, Brazil (CAPES) – Finance Code 001.

REFERENCES

[1] A. Memedi and F. Dressler, "Vehicular visible light communications: A survey," *IEEE Commun. Surveys Tutor.*, vol. 23, pp. 161–181, 1st quart. 2021.

[2] L. C. Mathias, L. F. de Melo, and T. Abrao, "Modeling and mitigation of spectral crosstalk in OFDM WDM-VLC system," *Optics Commun.*, vol. 478, p. 126361, Jan. 2021.

[3] A. Kafizov, A. Elzanaty, and M.-S. Alouini, "Probabilistic shaping for color-shift keying: Spectral-efficient transceiver design and prototype," *IEEE Trans. Wirel. Commun.*, vol. 25, pp. 6020–6034, Oct. 2025.

[4] "IEEE standard for local and metropolitan area networks–Part 15.7: Short-range optical wireless communications," *IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011)*, pp. 1–407, Apr. 2019.

[5] V. N. Saxena, V. K. Dwivedi, and J. Gupta, "Machine learning in visible light communication system: A survey," *Wirel. Commun. Mob. Comput.*, vol. 2023, p. 3950657, Jul. 2023.

[6] R. M. Martins, L. C. Mathias, and T. Abrão, "3-D indoor visible light positioning using machine learning and hyperparameter tuning," *Journal of Network and Systems Management*, vol. 33, no. 4, p. 79, 2025.

[7] A. C. Gordillo, "CNN-based demodulation of color shift keying in screen camera communications," in *Proc. First Int. Conf. Comput. Vis. Mach. Intell. (I3CIS 20222)*, Jijel, Algeria, Nov. 2022, pp. 1–8.

[8] Y. Onodera, D. Hisano, K. Maruta, and Y. Nakayama, "First demonstration of 512-color shift keying signal demodulation using neural equalization for optical camera communication," in *Proc. Opt. Fiber Commun. Conf. Exhib. (OFC 2023)*, San Diego, USA, Mar. 2023, pp. 1–3.

[9] A. Sharma, K. Singh, V. Bhatia, and C.-P. Li, "CSK modulation scheme and DNN-based symbol detection in OSTAR-RIS-aided VLC systems," *IEEE Wirel. Commun. Lett.*, vol. 14, pp. 1–5, Apr. 2025.

[10] P. P. Játiva, C. A. Azurdia-Meza, D. Zabala-Blanco, C. A. Gutiérrez, I. Sánchez, F. R. Castillo-Soria, and F. Seguel, "Bit error probability of VLC systems in underground mining channels with imperfect CSI," *AEU Int. J. Electron. Commun.*, vol. 145, p. 154101, Aug. 2022.

[11] P. Palacios Játiva, I. Sánchez, I. Soto, C. A. Azurdia-Meza, D. Zabala-Blanco, M. Ijaz, and D. Plets, "A novel and adaptive angle diversity-based receiver for 6G underground mining VLC systems," *Entropy*, vol. 24, p. 1507, Oct. 2022.

[12] R. Becerra, C. A. Azurdia-Meza, P. Palacios Játiva, I. Soto, J. Sandoval, M. Ijaz, and D. F. Carrera, "A wavelength-dependent visible light communication channel model for underground environments and its performance using a color-shift keying modulation scheme," *Electron.*, vol. 12, p. 577, Jan. 2023.

[13] P. P. Játiva, C. A. Azurdia-Meza, M. R. Cañizares, I. Sánchez, F. Seguel, D. Zabala-Blanco, and D. F. Carrera, "Performance analysis of IEEE 802.15.7-based visible light communication systems in underground mine environments," *Photon. Netw. Commun.*, vol. 43, pp. 23–33, Feb. 2022.

[14] A. Yokoi and J. Son, "More description about CSK constellation," IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), Contribution to IEEE 802.15.7 TG-VLC, September 2010, submitted by Samsung Electronics Co., LTD.

[15] *Color GT-P02-XX*, Getian Group, 2015. [Online]. Available: https://img.gme.cz/files/eshop_data/eshop_data/4/518-122/dsh.518-122.1.pdf

[16] L. C. Mathias, A. R. C. e Souza, and T. A. ao, "Wavelength widths of optical filters for optimum SINR in WDM-VLC systems," *Appl. Opt.*, vol. 59, pp. 5615–5624, Jun. 2020. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-59-18-5615

[17] C.-D. Tong, X. Zheng, X. Huang, P.-X. Zeng, Q. Deng, Y.-Y. Chen, T.-Z. Wu, Y.-J. Lu, Z. Chen, and W.-J. Guo, "The impacts of optical crosstalk on the color gamut of mini-LED integrated matrix device," *IEEE Photon. Technol. Lett.*, vol. 36, pp. 957–960, Aug. 2024.

[18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[19] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.

[20] M. T. Brown and L. R. Wicker, "8 - discriminant analysis," in *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, H. E. Tinsley and S. D. Brown, Eds. San Diego: Academic Press, 2000, pp. 209–235.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[22] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed. Wiley Publishing, 2007.

[23] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold networks," 2025. [Online]. Available: https://arxiv.org/abs/2404.19756

[24] Q. Qiu, T. Zhu, H. Gong, L. Chen, and H. Ning, "Relu-kan: New kolmogorov-arnold networks that only need matrix addition, dot multiplication, and relu," 2024. [Online]. Available: https://arxiv.org/abs/2406.02075

[25] C. A. Gueymard, "The sun's total and spectral irradiance for solar energy applications and solar radiation models," *Solar Energy*, vol. 76, no. 4, pp. 423–453, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X03003967

APPENDIX C – DEEP UNFOLDED SCA JOINT RADAR-COMMUNICATION PRECODING

Deep Unfolded SCA for ISAC Non-Convex Precoding Design: Learning Adaptive Schedules

Journal Article

Submitted to the **IEEE Transactions on Wireless Communications (TWC)** in January 2026.

Manuscript ID: Paper-TW-Jan-26-0226.

Impact Factor	8.9 (2024 JCR)
Qualis Capes	A1
Submission Date	21 Jan. 2026
Status	R1

Deep Unfolding-Based SCA for ISAC Non-Convex Precoding Design: Learning Adaptive Schedules

Rafael Marasca Martins, Bruno Felipe Costa, Luis C. Mathias, and Taufik Abrão

***Abstract*—The design of dual-functional precoders for Integrated Sensing and Communication (ISAC) systems is challenging due to non-convex objectives and stringent real-time latency constraints. To bridge the gap between rigorous optimization and online feasibility, we propose Deep Unfolded SCA (DU-SCA), a deep learning framework unfolding a nested proximal Successive Convex Approximation (SCA) algorithm into a fixed-depth neural network. This architecture preserves the problem's geometry while replacing static hyperparameters with adaptive, jointly learned schedules. Furthermore, a differentiable residual update bypasses Armijo line-searches, ensuring deterministic online execution. Benchmarking against a tuned SCA baseline, an Semidefinite Relaxation (SDR) bound, and a plain Deep Neural Network (DNN) reveals three key advantages. First, unlike plain DNNs which struggle to reliably produce feasible precoders, DU-SCA achieves near-parity with the tuned classical SCA in average sensing performance at nominal operating points, with the classical baseline retaining a marginal advantage in average minimum Signal to Interference plus Noise Ratio (SINR) satisfaction, while DU-SCA operates at a consistently lower mean transmit power.Second, DU-SCA demonstrates high robustness under strictly constrained computational budgets and overloaded user scenarios. At a limited budget of just 25 iterations, DU-SCA achieves over 97% feasibility, whereas the classical baseline drops below 5%. Finally, by replacing data-dependent iterative stopping criteria with a fixed-depth forward pass, DU-SCA delivers a consistent average per-sample speedup of approximately 1.8× over the tuned classical baseline, providing a deterministic and practical solution for high-performance ISAC precoding within strict hardware budgets.**

***Index Terms*—Integrated sensing and communication (ISAC), Deep Unfolding (DU), Successive Convex Approximation (SCA), Machine Learning (ML), Joint Radar-Communication (JRC), Precoding**

I. Introduction

Integrated Sensing and Communication (ISAC) has emerged as a critical enabler for next-generation networks, offering a pathway to efficient spectrum utilization by unifying radar and data transmission on a single hardware platform [1]–[3]. Early works established the benefits of these joint designs, demonstrating that sharing spectral and hardware resources can effectively enhance system performance while alleviating congestion [4]–[6]. However, achieving optimal performance in such dual-function systems introduces complex signal processing challenges, primarily due to the conflicting radar and communication objectives [1], [3]–[5]. To resolve this trade-off, advanced beamforming techniques have been developed to synthesize signals that match desired radar beampatterns while simultaneously satisfying strict communication Quality of Service (QoS) requirements [7]–[9].

More recent designs move from beampattern matching to direct Cramér-Rao Bound (CRB) minimization, which rigorously quantifies sensing accuracy and yields superior target-parameter estimation [3], [6], [10], [11]. Because the sensing objectives and communication constraints are tightly coupled, the resulting problems are inherently non-convex. This complexity was addressed by a robust Successive Convex Approximation (SCA) framework [10], [11] that iteratively linearizes the non-convex constraints into tractable Second-Order Cone Program (SOCP)s, guaranteeing convergence to a stationary point [11]. This core methodology extends effectively to full-duplex ISAC [12] and wideband hybrid beamforming [13]. In practice, it achieves near-Semidefinite Relaxation (SDR) performance, reducing the CRB by 10–15 dB over traditional methods in under 20 iterations [11]. Moreover, the extended frameworks unlock significant system gains, including sum-rate improvements of up to 70% with fewer radio frequency (RF) chains [13] and substantial efficiency boosts over half-duplex alternatives [12].

Although these optimization-based approaches can, in principle, deliver near-optimal performance, their computational complexity is often prohibitive for real-time operation in fast-varying ISAC scenarios, especially when wideband channels, large antenna arrays, or hybrid analog-digital architectures are involved [14]–[17]. Purely data-driven alternatives are not a satisfying response: while machine learning has been widely explored to enhance telecommunications through data-driven inference, standard deep learning techniques often lack theoretical performance guarantees and interpretability [18], [19]. Operating primarily as black-box models, these architectures rely on generic topologies (e.g., Convolutional Neural Network (CNN)s or Deep Neural Network (DNN)s) that ignore the underlying physical mechanisms of the signal processing task [16], [18]–[20]. Consequently, they require enormous volumes of labeled training data to approximate the complex channel-to-precoder mapping from scratch and often struggle with generalization outside their training distribution [16], [19], [21].

To bridge the gap between reliable optimization and rapid inference, Deep Unfolding (DU) has emerged as a transformative solution for ISAC systems [16], [22], [23]. Distinct from black-box approaches, DU does not attempt to learn the mapping functionality. Instead, it unfolds the iterations of a rigorous model-based algorithm to construct a trainable deep

This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, and in part by Brazil's National Council for Scientific and Technological Development (CNPq) under Grants 314618/2023-6 and 442945/2023-0.

R. M. Martins, B. F. Costa, and T. Abrão are with the Department of Electrical Engineering (DEEL). State University of Londrina (UEL). Po.Box 10.011; Londrina 86057-970, PR - Brazil. Email: rafael.marasca@uel.br, and taufik@uel.br.

L. C. Mathias is with the Computer Engineering Undergraduate Program (COENC). Federal Technological University of Paraná (UTFPR). Toledo 85902-490, PR - Brazil. Email: mathias@utfpr.edu.br.

architecture, thereby inheriting the principled structure and domain knowledge of the original optimization problem as a strong inductive bias [19], [20], [22], [24]. By restricting the learning process to a small set of algorithmic hyperparameters (e.g., step sizes and penalty weights) rather than millions of unconstrained weights, DU achieves superior data efficiency, interpretability, and convergence speed compared to both purely model-based and purely data-driven alternatives [15], [16].

A. Related Works

The existing literature relevant to this study can be broadly categorized into two tracks: model-based optimization for CRB-constrained ISAC beamforming, and DU techniques for low-latency wireless inference. The proposed architecture bridges these domains.

Within the model-based domain, various low-complexity solvers have advanced the CRB-minimization framework of [10], [11]. Recent methods include dual-domain reformulations [25], [26], augmented-Lagrangian penalty tuning [27], structural optimization algorithms [28], and WMMSE-based tradeoff schemes [29]–[31]. While these works advance asymptotic solution quality, their online execution relies on iterative, data-dependent stopping criteria, rendering them incompatible with the sub-millisecond latency budgets of real-time ISAC.

Conversely, DU has emerged as a robust solution for non-convex wireless design. It has been successfully applied to Intelligent Reflecting Surface (IRS) beamforming [32], power allocation [33], Hybrid Beamforming (HBF) projection [14], and Alternating Direction Method of Multipliers (ADMM)-based approximations [23], [34]. Furthermore, [32] demonstrated that stochastic SCA can be unfolded for IRS-assisted full-duplex systems. However, none of these architectures target the specific nested proximal-SCA pipeline characteristic of CRB-constrained ISAC.

Consequently, a critical methodological gap remains: model-based ISAC lacks deterministic-depth online solvers, while existing unfolded ISAC designs inherit online line-search behavior. The present work addresses this by unfolding the complete nested proximal-SCA solver into a fixed-depth architecture featuring jointly learned schedules. By eliminating the online line-search, the proposed method guarantees deterministic latency and is benchmarked against both model-based and black-box baselines to isolate its structural advantages.

B. Motivation and Contributions

Building on the CRB-minimization framework of [10], [11], we target the joint design of communication and sensing precoders to minimize the sensing CRB subject to strict per-user QoS and power constraints. While the traditional proximal-SCA pipeline is theoretically robust, its online application is severely restricted by heavy matrix inversions, serial Armijo line-search bottlenecks, and static hyperparameters. To bridge the gap between mathematically rigorous optimization and real-time execution, we propose Deep Unfolded SCA (DU-SCA). The main contributions of this work are summarized as follows:

- **Comprehensive Nested Architecture and Methodological Blueprint:** We propose DU-SCA, a deep-unfolded framework that maps the complete nested proximal-SCA procedure,including the outer surrogate-refinement loop and the inner gradient-based subproblem solver, into a fixed-depth computational graph. Furthermore, we provide an in-depth methodological roadmap detailing this implementation, offering a highly reproducible blueprint for translating complex, nested optimization geometries into efficient neural architectures.
- **Sub-Millisecond Deterministic Latency:** We bypass computationally expensive, data-dependent Armijo line-searches by introducing adaptive, layer-wise learnable schedules. By co-optimizing inner step sizes, momentum, proximal weights, and penalty parameters, the network autonomously learns a highly efficient optimization policy. This successfully eliminates serial bottlenecks, delivering the guaranteed, sub-millisecond deterministic execution times required for practical ISAC deployment.
- **Superiority Under Constrained Computational Budgets:** We demonstrate that DU-SCA makes a highly compelling case for practical application by bridging the gap between theoretical performance and real-time feasibility. At nominal operating points, DU-SCA matches the average CRB of a finely tuned classical SCA baseline. Crucially, the primary advantage of DU-SCA manifests under severely truncated iteration budgets. By achieving 97.7% constraint satisfaction at just 25 iterations—where the classical baseline catastrophically fails at 4.86%—DU-SCA demonstrates exceptionally rapid convergence to feasible solutions, allowing system designers to aggressively cap iteration limits to meet sub-millisecond real-time deadlines.”
- **Rigorous Validation of Model-Driven Priors:** We benchmark DU-SCA against a comprehensive suite of references, including a Convex Relaxation (CR)-based quality bound and a plain DNN trained on the identical unsupervised loss. This structural comparison isolates the value of the proximal-SCA inductive prior, proving that while generic black-box models fail to reliably produce feasible precoders, DU-SCA successfully guarantees mathematical robustness without sacrificing online speed.

II. System Model

We consider a downlink ISAC system where a dual-functional Base Station (BS) is equipped with N_t transmit antennas and $N_r \geq N_t$ receive antennas. The BS serves K single-antenna communication users in a Multi User (MU)-Multiple-Input Single-Output (MISO) scenario while simultaneously performing monostatic radar sensing of the environment, as illustrated in Fig. 1.

A. Communication Signal Model

The dual-functional waveform transmitted by the BS is denoted as $\mathbf{X} \in \mathbb{C}^{N_t \times L_{\text{frame}}}$, where L_{frame} represents the frame length. The transmission satisfies a total power constraint

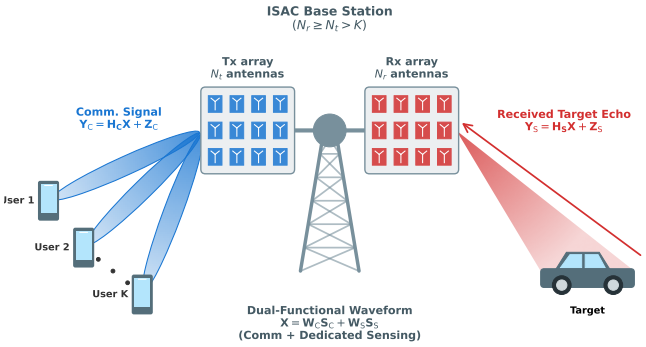


Fig. 1: Illustration of the joint radar-communication downlink (DL) system.

$\|\mathbf{X}\|_F^2 \leq P_{\mathrm{T}}$, with P_{T} denoting the maximum transmit power budget.

The signal received by the set of K communication users is modeled as [10], [11]:

$$\mathbf{Y}_{\mathrm{C}} = \mathbf{H}_{\mathrm{C}}\mathbf{X} + \mathbf{Z}_{\mathrm{C}}, \tag{1}$$

where $\mathbf{H}_{\mathrm{C}} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]^H \in \mathbb{C}^{K\times N_t}$ denotes the aggregate downlink communication channel matrix, assumed to be perfectly known at the BS. The individual downlink channels follow a Rician fading model, given by $\mathbf{h}_k = \sqrt{\kappa/(\kappa+1)}\,\bar{\mathbf{h}}_k + \sqrt{1/(\kappa+1)}\,\tilde{\mathbf{h}}_k$ for all $k \in \mathcal{K}$. In this formulation, κ is the Rician K-factor, $\bar{\mathbf{h}}_k$ is the deterministic line-of-sight component, and $\tilde{\mathbf{h}}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_t})$ is the scattered component. The special case of $\kappa = 0$ recovers standard Rayleigh fading. The noise component $\mathbf{Z}_{\mathrm{C}} \in \mathbb{C}^{K\times L}$ represents Additive White Gaussian Noise (AWGN) with elements independently drawn from $\mathcal{CN}(0, \sigma_{\mathrm{C}}^2)$.

B. Sensing Signal Model

Concurrently, the echo signal reflected from the target and captured by the BS sensing receiver can be expressed as [10]:

$$\mathbf{Y}_{\mathrm{S}} = \mathbf{H}_{\mathrm{S}}\mathbf{X} + \mathbf{Z}_{\mathrm{S}}, \tag{2}$$

where $\mathbf{H}_{\mathrm{S}} \in \mathbb{C}^{N_r\times N_t}$ denotes the sensing channel (or Target Response Matrix) to be estimated. The additive noise $\mathbf{Z}_{\mathrm{S}} \in \mathbb{C}^{N_r\times L}$ contains entries distributed as $\mathcal{CN}(0, \sigma_{\mathrm{S}}^2)$.[1]

Following the framework in [10], [11], we treat the target response matrix $\mathbf{H}_{\mathrm{S}}$ under a *non-parametric* sensing model. Rather than assuming the environment consists of a few discrete point targets characterized by specific physical parameters (such as angles of arrival or reflection coefficients), $\mathbf{H}_{\mathrm{S}}$ is modeled as an arbitrary, unknown deterministic matrix. This approach corresponds to sensing an extended target or a complex scattering environment. Consequently, the fundamental limits of estimating this response depend entirely on the spatial covariance of the transmitted signal, allowing us to optimize the precoders without requiring prior knowledge of a specific target geometry.

[1]Following standard assumptions in full-duplex ISAC systems, the self-interference from the transmit array is assumed to be perfectly suppressed via isolation and cancellation techniques [2], [12].

C. Radar Estimation Performance

We assume the BS employs the Maximum Likelihood Estimator (MLE) to recover $\mathbf{H}_{\mathrm{S}}$ from the noisy observation $\mathbf{Y}_{\mathrm{S}}$. The MLE is given by [2], [10], [11]:

$$\hat{\mathbf{H}}_{\mathrm{S}} = \mathbf{Y}_{\mathrm{S}}\mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}. \tag{3}$$

Considering that the received signal is corrupted exclusively by AWGN, the Mean Squared Error (MSE) of this estimator coincides with the CRB for the estimation of $\mathbf{H}_{\mathrm{S}}$, which is expressed as [2], [10], [11]:

$$\mathrm{CRB}(\mathbf{X}) = \mathbb{E}\left\{\|\mathbf{H}_{\mathrm{S}} - \hat{\mathbf{H}}_{\mathrm{S}}\|_F^2\right\} = \frac{\sigma_{\mathrm{S}}^2 N_r}{L_{\mathrm{frame}}}\,\mathrm{tr}\left(\mathbf{R}_{\mathrm{X}}^{-1}\right), \tag{4}$$

where $\mathbf{R}_{\mathrm{X}} = \frac{1}{L_{\mathrm{frame}}}\mathbf{X}\mathbf{X}^H$ is the sample covariance matrix of the transmitted signal.

Physically, the CRB characterizes the minimum achievable variance of any unbiased estimator and serves as a fundamental performance metric for sensing. In radar-centric ISAC systems, reducing the CRB improves the accuracy of target response estimation, which is closely linked to the structure of the transmit signal covariance matrix. In particular, a well-conditioned spatial covariance enhances scene illumination by distributing energy across spatial directions, thereby improving the quality of the received echoes and reducing estimation error.

A key implication of (4) is that effective sensing requires the transmit covariance ($\mathbf{R}_{\mathrm{X}}$) to be full rank; otherwise, the Fisher Information Matrix becomes singular and reliable estimation is not possible [2]. This condition directly constrains the structure of the transmitted signal.

However, this requirement conflicts with conventional multi-user MISO transmission. In standard MU-MISO systems, the base station transmits K independent data streams, which leads to $\mathrm{rank}(\mathbf{X}) \leq K$. In typical massive Multiple-Input Multiple-Output (MIMO) regimes where $N_t > K$, this results in a rank-deficient transmit covariance, limiting sensing performance under conventional designs [2], [8].

D. Precoding Design with Dedicated Sensing Signals

To overcome the rank deficiency indicated in the last section, we adopt the approach of introducing $N_t - K$ auxiliary signals dedicated to sensing, as in [2], [8], [10], [11]. These auxiliary signals contain no useful communication information and are designed to be orthogonal to the data streams, ensuring that $\mathbf{X}$ achieves full rank without interfering with the communication users.

The transmitted waveform is structured as a superposition of communication precoded data and dedicated sensing signals [2], [11]:

$$\mathbf{X} = \mathbf{W}_{\mathrm{C}}\mathbf{S}_{\mathrm{C}} + \mathbf{W}_{\mathrm{S}}\mathbf{S}_{\mathrm{S}}, \tag{5}$$

where $\mathbf{W}_{\mathrm{C}} \in \mathbb{C}^{N_t\times K}$ and $\mathbf{W}_{\mathrm{S}} \in \mathbb{C}^{N_t\times(N_t-K)}$ are the communication and sensing precoding matrices, respectively. The symbol matrices $\mathbf{S}_{\mathrm{C}} \in \mathbb{C}^{K\times L}$ and $\mathbf{S}_{\mathrm{S}} \in \mathbb{C}^{(N_t-K)\times L}$ are assumed to be uncorrelated with unit power, such that $\mathbb{E}[\mathbf{S}_{\mathrm{C}}\mathbf{S}_{\mathrm{C}}^H] = \mathbf{I}_K$, $\mathbb{E}[\mathbf{S}_{\mathrm{S}}\mathbf{S}_{\mathrm{S}}^H] = \mathbf{I}_{N_t-K}$, and $\mathbb{E}[\mathbf{S}_{\mathrm{C}}\mathbf{S}_{\mathrm{S}}^H] = \mathbf{0}$. This structure ensures that $\mathbf{X}$ attains full rank (N_t), thereby minimizing the estimation error bound.

E. Impact on Communication SINR

While the dedicated sensing signals $\mathbf{W}_{\mathrm{S}}\mathbf{S}_{\mathrm{S}}$ improve radar performance, they act as interference to the communication users. The resulting Signal to Interference plus Noise Ratio (SINR) for the k-th user is given by [2], [8]–[10]:

$$\gamma_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\underbrace{\sum_{i \neq k} |\mathbf{h}_k^H \mathbf{w}_i|^2}_{\text{MUI}} + \underbrace{\mathbf{h}_k^H \mathbf{W}_{\mathrm{S}} \mathbf{W}_{\mathrm{S}}^H \mathbf{h}_k}_{\text{Sensing Interference}} + \sigma_{\mathrm{C}}^2}, \quad \forall k \in \mathcal{K}. \tag{6}$$

where $\mathcal{K} = \{1, \ldots, K\}$ denotes the set of communication users. Here, $\mathbf{w}_k$ denotes the k-th column of $\mathbf{W}_{\mathrm{C}}$, corresponding to the precoder assigned to user k, while $\mathbf{w}_i$ denotes the precoder associated with user $i \neq k$ and therefore contributes to multi-user interference. The denominator comprises the Multi-User Interference (MUI), the sensing-induced interference, and the receiver noise. The optimization challenge lies in designing $\mathbf{W}_{\mathrm{C}}$ and $\mathbf{W}_{\mathrm{S}}$ to minimize the CRB while managing this interference to satisfy the per-user SINR requirements.

III. Optimization Problem Formulation

The primary objective of the proposed Joint Radar-Communication (JRC) system is to jointly design the communication and sensing precoders, denoted by the dual-functional matrix $\mathbf{W} = [\mathbf{W}_{\mathrm{C}}, \mathbf{W}_{\mathrm{S}}] \in \mathbb{C}^{N_t \times N_t}$, to optimize sensing performance while guaranteeing the QoS for communication users.

A. Problem Statement

As established in [10], under the standard Gaussian linear model, the MSE of the unbiased MLE coincides with the CRB. Consequently, minimizing the estimation error is equivalent to minimizing the trace of the inverse sample covariance matrix. Assuming the frame length L is sufficiently large, the sample covariance matrix approximates the statistical covariance, i.e., $\mathbf{R}_{\mathrm{X}} \approx \mathbf{W}\mathbf{W}^H$ [10], [11].

The optimization problem is then formulated to minimize the CRB metric defined in (4), subject to strict per-user SINR constraints and a total transmit power budget [10], [11]:

$$\min_{\mathbf{W}} \quad \mathrm{CRB}(\mathbf{W}) \approx \mathcal{L}_{\mathrm{CRB}}(\mathbf{W}) = \mathrm{tr}\left((\mathbf{W}\mathbf{W}^H)^{-1}\right) \tag{7a}$$

$$\text{s.t.} \quad \gamma_k \geq \Gamma_k, \quad \forall k \in \mathcal{K}, \tag{7b}$$

$$\|\mathbf{W}\|_F^2 \leq P_T, \tag{7c}$$

where P_T represents the maximum transmit power budget, and Γ_k is the required SINR threshold for the k-th user. The objective function (7a) corresponds directly to the minimization of the CRB in (4), ensuring the optimal lower bound on the radar estimator variance.

B. Differentiable Cost Function Formulation

The hard, non-convex constraints in (7a)–(7c) are incompatible with direct gradient-based learning [8], [11], [14]. To enable DU, we relax them via a squared Rectified Linear Unit (ReLU) penalty, $P(z) = ([z]^+)^2$, which is continuously differentiable (C^1), imposes zero cost on satisfied constraints, and provides stronger gradient feedback for large violations than a linear penalty [35]–[37].

Following the constrained optimization framework, the resulting unconstrained loss function is formulated as:

$$\mathcal{L}_{\text{total}}(\mathbf{W}) = \mathcal{L}_{\mathrm{CRB}}(\mathbf{W}) + \lambda_{\mathrm{p}} \mathcal{L}_{\mathrm{pwr}}(\mathbf{W}) + \lambda_{\mathrm{s}} \mathcal{L}_{\mathrm{sinr}}(\mathbf{W}), \tag{8}$$

where λ_{p}, and λ_{s} are non-negative hyperparameters balancing the trade-off between estimation accuracy and constraint satisfaction. In this composite objective, $\mathcal{L}_{\mathrm{CRB}}$ drives the primary sensing task by minimizing the estimation error variance. Conversely, $\mathcal{L}_{\mathrm{pwr}}$ and $\mathcal{L}_{\mathrm{sinr}}$ function as penalty terms that police the feasibility of the solution; they impose high costs on the network whenever the physical power budget or the communication QoS targets are violated.

The power penalty term, $\mathcal{L}_{\mathrm{pwr}}$, enforces the total energy budget by penalizing the squared Frobenius norm of the precoding matrix whenever it exceeds the threshold P_T:

$$\mathcal{L}_{\mathrm{pwr}}(\mathbf{W}) = \left[\mathrm{ReLU}\left(\|\mathbf{W}\|_F^2 - P_T\right)\right]^2. \tag{9}$$

The SINR penalty term, $\mathcal{L}_{\mathrm{sinr}}$, addresses the non-convexity of the minimum SINR requirements. Following the approach in [10], [11], we reformulate the constraints into a Second-Order Cone (SOC) form. By exploiting the phase ambiguity of the precoders to enforce a real-valued, non-negative useful signal component, the violation can be penalized as [10], [11]:

$$\mathcal{L}_{\mathrm{sinr}}(\mathbf{W}) = \sum_{k=1}^{K} \left[\mathrm{ReLU}\left(\sqrt{\Gamma_k} \|\tilde{\mathbf{h}}_k\|_2 - \sqrt{1+\Gamma_k} \Re\{\mathbf{h}_k^H \mathbf{w}_k\} \right) \right]^2, \tag{10}$$

where $\|\tilde{\mathbf{h}}_k\|_2 = \|[\mathbf{h}_k^H \mathbf{W}, \sigma_C]\|_2$ represents the Euclidean norm of the augmented interference-plus-noise vector.

IV. Successive Convex Approximation (SCA)

The optimization problem formulated in (7) involves minimizing the objective function $\mathcal{L}_{\mathrm{CRB}}(\mathbf{W}) = \mathrm{tr}((\mathbf{W}\mathbf{W}^H)^{-1})$, which is non-convex with respect to $\mathbf{W}$ [10], [11]. To address this, we employ a proximal-SCA strategy [11], [38]. This iterative method approximates the non-convex landscape with a sequence of strongly convex subproblems, ensuring convergence to a stationary point through rigorous line search procedures [10], [39], [40].

A. Linearization and Gradient Calculation

At each iteration t, we construct a convex surrogate by linearizing the objective $\mathcal{L}_{\mathrm{CRB}}(\mathbf{W})$ around the current operating point $\mathbf{W}^{(t-1)}$. To simplify notation, let us define the gradient at the current iteration as $\mathbf{G}^{(t-1)} \triangleq \nabla \mathcal{L}_{\mathrm{CRB}}(\mathbf{W}^{(t-1)})$. The gradient is derived using complex matrix calculus as [10], [11]:

$$\begin{aligned} \mathbf{G}^{(t-1)} &= -\left(\mathbf{W}^{(t-1)}(\mathbf{W}^{(t-1)})^H\right)^{-1} \\ &\quad \times \left(\mathbf{W}^{(t-1)}(\mathbf{W}^{(t-1)})^H\right)^{-H} \mathbf{W}^{(t-1)} \\ &= -(\mathbf{R}_{\mathrm{X}}^{-1})^2 \mathbf{W}^{(t-1)}. \end{aligned} \tag{11}$$

B. Convex Subproblem Formulation

To ensure the strict convexity of the subproblem and stabilize the updates, we introduce a proximal regularization term $\frac{\rho}{2}\|\mathbf{W}-\mathbf{W}^{(t-1)}\|_F^2$ [38], [41]. This term acts as a soft trust-region constraint, penalizing deviations from the point where the linearization is valid [41].

The convex subproblem at iteration t minimizes the following surrogate loss function:

$$\begin{aligned}\hat{\mathcal{L}}^{(t)}(\mathbf{W}) = \Re\left\{\mathrm{tr}\left((\mathbf{G}^{(t-1)})^H(\mathbf{W}-\mathbf{W}^{(t-1)})\right)\right\} \\ + \frac{\rho}{2}\|\mathbf{W}-\mathbf{W}^{(t-1)}\|_F^2 \\ + \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{pwr}}(\mathbf{W}) + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{sinr}}(\mathbf{W}).\end{aligned} \tag{12}$$

C. Optimization via Nesterov Accelerated Gradient Descent

To efficiently solve the strongly convex subproblem in (12), we employ Nesterov Accelerated Gradient Descent (NAGD) with the velocity-based formulation [42], which is well-suited for deep learning architectures. This method evaluates gradients at a look-ahead position using momentum, enabling faster course correction and improved convergence on high-curvature landscapes.

Let $\mathbf{V}^{(l)}$ denote the velocity matrix (initialized to zero) and $\mathbf{W}^{(l)}$ the optimization variable at iteration l, controlled by momentum coefficient μ and step size η. The look-ahead position is:

$$\tilde{\mathbf{W}}^{(l)} = \mathbf{W}^{(l)} + \mu\mathbf{V}^{(l)}. \tag{13}$$

Next, we evaluate the gradient of the surrogate loss at this extrapolated position, denoted as $\mathbf{G}^{(l)} = \nabla_{\mathbf{W}}\hat{\mathcal{L}}^{(t)}(\tilde{\mathbf{W}}^{(l)})$. To ensure training stability, we apply per-sample gradient clipping [43]:

$$\bar{\mathbf{g}}_i^{(l)} = \min\left(1, \frac{C}{\|\mathbf{g}_i^{(l)}\|_2+\epsilon}\right)\mathbf{g}_i^{(l)}, \tag{14}$$

where C is the maximum gradient norm and ϵ is a small constant for numerical stability. Finally, we update the velocity and the variable using the step size η explicitly:

$$\mathbf{V}^{(l+1)} = \mu\mathbf{V}^{(l)} - \eta\bar{\mathbf{G}}^{(l)}, \tag{15a}$$

$$\mathbf{W}^{(l+1)} = \mathbf{W}^{(l)} + \mathbf{V}^{(l+1)}. \tag{15b}$$

This process repeats for a fixed number of iterations (L) or until the gradient norm falls below a specified tolerance.

D. Step Size Selection via Armijo Rule

Let $\mathbf{W}^\star$ denote the optimal solution to the subproblem (12). In the SCA framework, this solution defines a descent direction $\mathbf{D}^{(t)} = \mathbf{W}^\star - \mathbf{W}^{(t-1)}$ [11]. While the direction $\mathbf{D}^{(t)}$ guarantees descent in the surrogate model, choosing the outer update step too large or too small can still prevent a sufficient decrease in the original non-convex objective, leading to divergence or stagnation [35], [36]. To promote global convergence, the accepted outer step size is denoted by $\alpha \in (0,1]$ and is selected by Armijo backtracking [35], [36]. This line-search rule guarantees a sufficient decrease relative to the directional derivative and therefore promotes monotonic progress toward a stationary point even in non-convex landscapes [35], [44].

The algorithm searches over the candidate set $\alpha \in \{\beta_{\mathrm{arm}}^0, \beta_{\mathrm{arm}}^1, \beta_{\mathrm{arm}}^2, \dots\}$, where $\beta_{\mathrm{arm}} \in (0,1)$ is the Armijo reduction factor that shrinks the candidate outer step size, and accepts the largest α satisfying the Armijo condition [36]:

$$\begin{aligned}\mathcal{L}_{\mathrm{total}}(\mathbf{W}^{(t-1)} + \alpha\mathbf{D}^{(t)}) \le \mathcal{L}_{\mathrm{total}}(\mathbf{W}^{(t-1)}) \\ + c\cdot\alpha\cdot\Re\left\{\left\langle\nabla\mathcal{L}_{\mathrm{total}}^{(t-1)}, \mathbf{D}^{(t)}\right\rangle\right\},\end{aligned} \tag{16}$$

where $c \in (0,1)$ is a control parameter and $\nabla\mathcal{L}_{\mathrm{total}}^{(t-1)}$ is the gradient of the total loss at the previous iterate. Once α is found, the variables are updated as [10], [11]:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \alpha\mathbf{D}^{(t)}. \tag{17}$$

V. Proposed Deep Unfolding-Based SCA Precoding

To circumvent the computational bottlenecks of the traditional SCA solver, we propose a model-driven DU-SCA framework. As illustrated in Fig. 2, the iterative process of the proximal-SCA algorithm is mapped onto a fixed-depth computational graph comprising T cascaded layers, where each layer corresponds to a single iteration of the original algorithm. This transformation effectively converts iterative optimization into a feed-forward architecture, fusing the theoretical interpretability of convex optimization with the inference efficiency of deep learning.

The proposed DU framework explicitly embeds the analytical structure and constraints of (8) as inductive priors. Within this architecture, algorithmic hyperparameters, specifically step sizes, momentum coefficients, and penalty weights, are parameterized as trainable variables. Through offline training, the network learns an optimized update trajectory tailored to the statistical distribution of the channel data [21], [24]. This enables the network to approximate the stationary solution of the SCA problem with a substantially reduced number of layers [21], [24]. By offloading the high-complexity adaptation to the offline training phase, the DU framework facilitates real-time inference with negligible online computational cost, thereby satisfying the stringent latency requirements of modern ISAC applications [24].

A. Adaptive Weighting Mechanism

As discussed above, one critical limitation of classical SCA is its reliance on static hyperparameters, which typically require tedious manual tuning via extensive simulations [22]. This manual calibration is often computationally prohibitive because standard solvers lack optimal parameter selection rules for finite-iteration regimes, forcing reliance on heuristics or pessimistic bounds that do not generalize across varying channel conditions [22].

From an optimization perspective, penalty-based objective functions are inherently sensitive to the weighting parameters [18], [22]. As established in nonlinear programming theory, an inappropriate choice of penalties yields severe consequences: insufficient penalty weights may fail to force the solution into the feasible set, resulting in physically invalid precoders, whereas excessively large weights induce ill-conditioning that causes the optimization trajectory to oscillate, significantly prolonging convergence time [18], [35], [36].

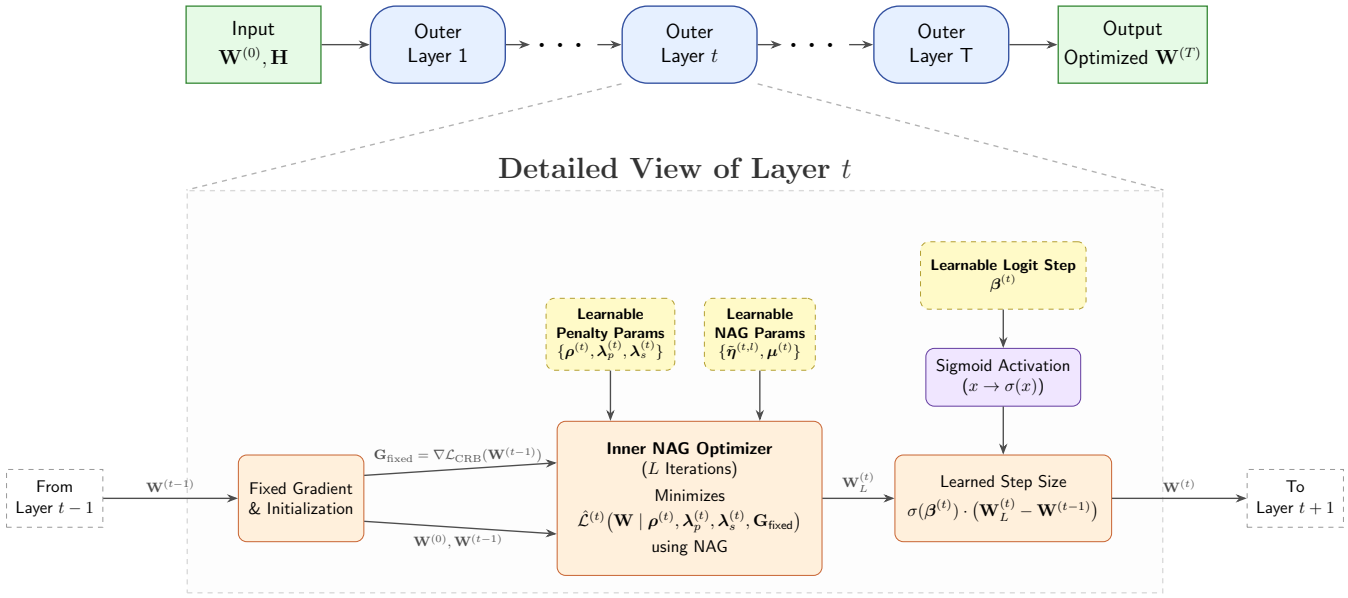


Fig. 2: Topology of the proposed deep unfolding-based SCA (DU-SCA) network. The architecture consists of T outer layers, where each layer t performs a convex approximation step refined by an inner optimization loop with learnable parameters.

In the context of ISAC, this sensitivity is further exacerbated, where deviations from the optimal operating point can severely degrade the delicate trade-off between sum rate and sensing accuracy [14], [18].

To overcome these stability issues, our DU-SCA architecture employs an adaptive weighting mechanism, where these hyperparameters are replaced by learnable variables that adaptively vary across layers. To ensure physical validity (i.e., weights must be strictly non-negative), we parameterize them in the log-domain [45]. Let $\boldsymbol{\Theta}^{(t)} = \{\tilde{\rho}^{(t)}, \tilde{\lambda}_{\text{p}}^{(t)}, \tilde{\lambda}_{\text{s}}^{(t)}\}$ denote the set of learnable hyperparameters for layer t. The actual physical weights used in the loss function are obtained via an exponential mapping:

$$\rho^{(t)} = e^{\tilde{\rho}^{(t)}}, \quad \lambda_{\text{p}}^{(t)} = e^{\tilde{\lambda}_{\text{p}}^{(t)}}, \quad \lambda_{\text{s}}^{(t)} = e^{\tilde{\lambda}_{\text{s}}^{(t)}}. \tag{18}$$

1) Role of Proximal Weight: The parameter $\rho^{(t)}$ governs the trust region of the update at layer t [38], [41]:

- **Conservative Updates (High $\rho^{(t)}$):** Enforces the solution to stay close to the linearization point $\mathbf{W}^{(t-1)}$, ensuring stability when the approximation is loose [38], [41].
- **Aggressive Updates (Low $\rho^{(t)}$):** Permits larger step sizes, accelerating convergence when the local convex approximation is reliable [38], [41].

2) Role of Penalty : These weights govern the critical trade-off between objective minimization and rigorous constraint satisfaction [22], [35], [46], [47]:

- **Relaxation Regime (Low Penalty Weights):** Lower penalty values effectively relax the hard constraint boundaries. This permits temporary violations, allowing the optimization trajectory to traverse infeasible regions of the landscape to escape shallow local minima [46], [47].
- **Enforcement Regime (High Penalty Weights):** Higher penalty values impose a prohibitive cost on infeasibility. This strict penalization forces the solution to converge within the feasible region, ensuring that the final output satisfies the physical SINR and power constraints [46], [47].

By learning the weights, the network can dynamically adjust its trajectory: taking aggressive steps to traverse the solution space quickly, and adopting a conservative approach to fine-tune the solution, possibly accelerating the convergence [22].

B. Inner Loop Implementation

Each outer layer t aims to solve the convex subproblem in (12). To achieve this efficiently, we unfold the solver into L iterations of NAGD [42]. Crucially, unlike classical implementations that rely on fixed or heuristically decaying schedules, our architecture parameterizes the solver with learnable control variables [22].

Specifically, for every inner iteration $l \in \{1, \ldots, L\}$ within outer layer t, we assign a unique step size $\eta^{(t,l)}$ and momentum coefficient $\mu^{(t,l)}$. These parameters are learned independently for each step of the unfolded architecture. This design allows the network to discover a highly dynamic, non-monotonic optimization schedule, customizing the acceleration profile not just for the global SCA procedure, but for the specific curvature characteristics of each intermediate subproblem.

To guarantee that the step sizes remain strictly positive, we parameterize them in the log-domain. Let $\tilde{\eta}^{(t,l)}$ be the learnable parameter stored in the network. The actual step size used in the update is given by:

$$\eta^{(t,l)} = e^{\tilde{\eta}^{(t,l)}}. \tag{19}$$

Similarly, the layer-wise momentum coefficient is constrained to $\mu^{(t)} \in (0, 1)$ via a sigmoid parameterization:

$$\mu^{(t)} = \sigma(\tilde{\mu}^{(t)}) = \frac{1}{1 + e^{-\tilde{\mu}^{(t)}}}. \tag{20}$$

C. Outer Loop Update

Upon completion of the L inner iterations, which yield the refined candidate solution $\mathbf{W}^\star$, the outer layer updates the global solution. In the classical SCA method, proposed by [10], [11], this update relies on backtracking line-search mechanisms (e.g., the Armijo rule) to guarantee monotonic descent. However, this process is computationally expensive, requiring repeated evaluations of the objective and constraints at each backtracking step, which creates a serial bottleneck ill-suited for real-time applications [16], [36].

To eliminate this overhead, our proposed DU-SCA framework replaces the iterative search with a residual learning strategy governed by a learned mixing parameter. This mechanism adaptively determines the outer update magnitude for each layer in a "one-shot" manner, balancing retention of prior information with integration of the refined candidate without costly iterative checks [22]. We introduce a learnable scalar $\beta^{(t)}$ to control this layer-wise outer step size via a sigmoid gating function:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \sigma(\beta^{(t)}) \left(\mathbf{W}^\star - \mathbf{W}^{(t-1)} \right). \tag{21}$$

where $\sigma(\cdot)$ denotes the sigmoid function, which constrains the update coefficient to the interval $(0, 1)$.

D. DU-SCA Training

The DU-SCA architecture is trained end-to-end using an unsupervised learning strategy. Unlike supervised approaches that require pre-computed optimal labels, our network directly minimizes the physical objective function derived from the problem formulation. The training loss $\mathcal{L}_{\text{total}}$, defined in (8), is computed as a weighted aggregate of the performance objective and constraint violations, where λ_{pwr} and λ_{sinr} are static hyperparameters utilized during training to heavily penalize infeasible solutions.

The optimization is performed using the Adaptive Moment Estimation (Adam) solver, which adapts the learning rates for individual parameters based on first and second-order moment estimates [48]. To ensure numerical stability, particularly given the recurrent nature of the unfolded layers, we employ gradient clipping to restrict the norm of the gradient vectors, preventing the exploding gradient problem common in deep equilibrium models [43]. Furthermore, we implement a learning rate annealing schedule, which systematically decays the learning rate by a fixed factor every N epochs to facilitate fine-grained convergence in the final training stages [49].

The complete forward pass of the proposed architecture is detailed in Algorithm 1.

Algorithm 1 Deep Unfolding-Based SCA Forward Pass

Require: Channel $\mathbf{H}$, Init $\mathbf{W}^{(0)}$, Layers T, Inner Steps L
Ensure: Optimized Precoder $\mathbf{W}^{(T)}$

for $t = 1$ to T **do**
  $\mathbf{G}_{\text{fixed}} \leftarrow \nabla \mathcal{L}_{\text{CRB}}(\mathbf{W}^{(t-1)})$
  Inner Init: $\mathbf{W}_0 \leftarrow \mathbf{W}^{(t-1)}$, $\mathbf{V} \leftarrow \mathbf{0}$
  // Map log-parameters to physical weights
  $\rho \leftarrow e^{\tilde{\rho}^{(t)}}, \quad \lambda_{\text{p}} \leftarrow e^{\tilde{\lambda}_{\text{p}}^{(t)}}, \quad \lambda_{\text{s}} \leftarrow e^{\tilde{\lambda}_{\text{s}}^{(t)}}$
  for $l = 1$ to L **do**
    // Nesterov Momentum
    $\mu \leftarrow \sigma(\mu^{(t)})$
    $\mathbf{W}_{\text{lookahead}} \leftarrow \mathbf{W}_{l-1} + \mu \cdot \mathbf{V}$
    // Compute Inner Gradients
    $\mathbf{g} \leftarrow \nabla_{\mathbf{W}} \hat{\mathcal{L}}^{(t)}(\mathbf{W}_{\text{lookahead}} \mid \rho, \lambda_{\text{p}}, \lambda_{\text{s}})$
    Clip $\|\mathbf{g}\|$ to G_{clip}
    // Update Velocity and Position
    $\eta \leftarrow e^{\tilde{\eta}_l^{(t)}}$
    $\mathbf{V} \leftarrow \mu \cdot \mathbf{V} - \eta \cdot \mathbf{g}$
    $\mathbf{W}_l \leftarrow \mathbf{W}_{l-1} + \mathbf{V}$
  end for
  $\mathbf{W}_{\text{star}} \leftarrow \mathbf{W}_L$
  // Outer Update with Learned Step
  $\beta \leftarrow \sigma(\beta^{(t)})$
  $\mathbf{W}^{(t)} \leftarrow \mathbf{W}^{(t-1)} + \beta \cdot (\mathbf{W}_{\text{star}} - \mathbf{W}^{(t-1)})$
end for
return $\mathbf{W}^{(T)}$

VI. Numerical Results

In this section, we empirically evaluate the proposed DU-SCA architecture against three reference solvers that comprehensively span the relevant design space: (i) a tuned classical SCA baseline representing the foundational proximal-SCA framework for ISAC established by [10], [11], executed under the same 10×10 iteration budget; (ii) a CR reference based on the SDR of the CRB-based ISAC problem [5], [10], which serves as a high-fidelity performance bound at the covariance level; and (iii) a plain black-box DNN trained with the same unsupervised loss on identical channel datasets, which isolates the value of the model-driven inductive bias. The presentation is organized to first establish the nominal-operating-point behavior of all four methods. Subsequently, we stress-test the the classical SCA and DU-SCA, across varying load, trade-off, iteration-budget, and channel-mismatch to provide an in-depth comparison of both methods.

A. Simulation Setup

Following the framework established in [10], [11], we consider a downlink MIMO-ISAC system featuring an $N_t = 16$ antenna transmitter and a co-located sensing receiver with $N_r = 20$ antennas. The frame length is fixed at $L_{\text{frame}} = 30$. The noise variance for both the communication users (σ_{C}^2) and the sensing receiver (σ_{S}^2) is normalized to 0 dBm (1 mW). The total transmit power budget is set to $P_T = 30$ dBm (1 W), yielding a nominal transmit Signal-to-Noise Ratio (SNR) of 30 dB. As defined in Section II, the communication links follow a Rician fading model governed by the K-factor $\kappa \in \{0, 1, 3, 5, 10\}$. The nominal training and evaluation datasets utilize $\kappa = 0$ (standard Rayleigh fading), while the broader κ sweep is deployed specifically for the cross-channel robustness analysis in Section VI-F.

To ensure computational fairness and preserve physical realism, the deployable finite-budget solvers (the classical SCA and the proposed DU-SCA) are strictly constrained to an identical $10 \times 10 = 100$-step latency budget. Furthermore, to prevent benchmarking the DU architecture against an artificially weak mathematical baseline, the classical SCA solver was rigorously optimized for this nominal operating point. Table I details the extensive grid search protocol utilized to select its static hyperparameters.

TABLE I: Classical SCA tuning protocol for the baseline scenario $(N_t, K, \Gamma) = (16, 10, 15\,\text{dB})$ under a fixed $10 \times 10 = 100$-step budget.

Parameter	**Selected**	**Grid**
Tuning set: 500 channels	*Evaluated:* 8640 configurations	
Inner step size η	5×10^{-3}	$\{10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}\}$
Momentum coefficient μ	0.95	$\{0.5, 0.7, 0.8, 0.9, 0.95\}$
Power penalty weight λ_{p}	10^5	$\{5 \times 10^3, 10^4, 5 \times 10^4, 8 \times 10^4, 10^5, 5\times10^5\}$
SINR penalty weight λ_{s}	5×10^4	(same grid as λ_{p})
Proximal weight ρ	5.0	$\{0.1, 0.5, 1.0, 5.0\}$
Armijo reduction β_{arm}	0.3	$\{0.3, 0.5, 0.7\}$
Budget study: outer/inner splits in $\{5, 10, 20, 50\}\times\{5, 10, 20, 50\}$		

TABLE II: Simulation Parameters and Algorithm Setup

Parameter	**Value / Range**
System Model & Baseline Scenario	
Transmit Antennas (N_t)	16
Receive Antennas (N_r)	20
Communication Users (K)	$\{4, 8, \mathbf{10}, 12, 14\}$
User SINR Threshold ($\Gamma_k = \Gamma$)	$\{5, 10, \mathbf{15}, 20, 25\}$ dB
Transmit Power Budget (P_T)	**30 dBm** (1.0 W)
Noise Powers ($\sigma_{\text{C}}^2, \sigma_{\text{S}}^2$)	**0 dBm** (1.0 mW)
Communication Channel Model	Rician fading, $\kappa \in \{0, 1, 3, 5, 10\}$
Frame Length (L_{frame})	30 time slots
Classical SCA Baseline (Optimized)	
Outer / Inner Iterations	10×10 (total 100)
Inner Step Size (η)	5×10^{-3}
Armijo Reduction (β_{arm})	0.3
Momentum Coefficient (μ)	0.95
Proximal Weight (ρ)	5.0
Penalty Weights ($\lambda_{\text{p}}, \lambda_{\text{s}}$)	$(10^5, 5 \times 10^4)$ (static)
DU-SCA (Proposed)	
Outer Layers (T)	10 (low latency)
Inner Steps per Layer (L)	10
Training Channels (N_{train})	1,000 (unsupervised)
Penalty Initialization ($\lambda_{\text{p}}^{(0)}, \lambda_{\text{s}}^{(0)}$)	Tuned-baseline values $(10^5, 5 \times 10^4)$
Learnable Parameters	$\eta^{(t,l)}, \mu^{(t)}, \rho^{(t)}, \lambda_{\text{p}}^{(t)}, \lambda_{\text{s}}^{(t)}, \beta^{(t)}$

Note: Bold values indicate the default simulation settings.

Table II summarizes the system parameters and algorithmic configurations. Unless otherwise specified, evaluations are conducted at the *nominal operating point* indicated by the bolded values: $K = 10$ communication users and a target SINR of $\Gamma = 15\,\text{dB}$. Consistent with [11], this configuration enforces a spatially loaded regime (load factor $K/N_t \approx 0.6$) subject to strict QoS requirements, rigorously stressing the interference management capabilities of the beamformers.

Beyond the tuned classical SCA baseline, we include two additional benchmarks to separate model-based from learning-based performance. The first is a CR based on lifting $\mathbf{R}_{\text{X}} = \mathbf{W}\mathbf{W}^H \succeq \mathbf{0}$, following the SDR formulation used for CRB-based ISAC precoding in [5], [10]. The resulting convex Semidefinite Program (SDP) provides a covariance-level quality bound; a rank-one communication precoder is recovered by Gaussian randomization with $R = 8$ trials, and a sensing precoder is synthesized from the residual covariance to keep $\mathbf{R}_{\text{X}}$ full rank. The SDP is solved by Clarabel with tolerance $\epsilon = 10^{-4}$ and up to 5000 iterations. This benchmark therefore provides a high-quality sensing reference, while its runtime highlights why lift-and-randomize schemes are not viable under sub-millisecond latency.

The second benchmark is a black-box DNN included to isolate the value of the model-driven inductive bias. It uses a residual architecture with $B = 4$ hidden blocks of width 1024, the same structured output head, and the same physical warm starts and power/nullspace projections as the proposed solver. At the nominal setting $(N_t, K) = (16, 10)$, it has approximately 9.2×10^6 trainable parameters, versus only $T(L + 5) = 150$ scalar parameters in DU-SCA. It is trained unsupervised on the same 1000-channel dataset with the same loss (8), so the comparison isolates architecture rather than supervision.

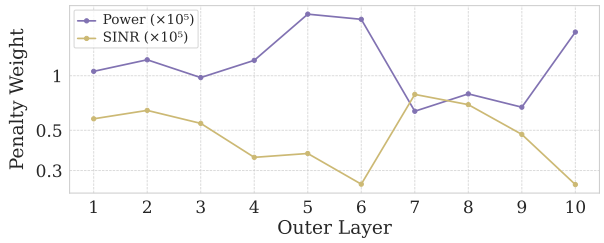
B. Training Convergence and Learned Schedules

The proposed DU-SCA architecture replaces manual hyperparameter tuning with an adaptive learning mechanism, optimizing a compact set of scalar parameters end-to-end. The offline training protocol utilizes $N_{\text{train}} = 1000$ unsupervised Rician channel realizations with $\kappa = 0$ (standard Rayleigh fading) to minimize the unified loss (8). The offline training process yields a fully autonomous optimization policy tailored to the CRB/SINR landscape. The true value of this DU approach is revealed by examining the layer-wise evolution of the learned hyperparameters, which autonomously converge to distinct, theoretically grounded schedules.

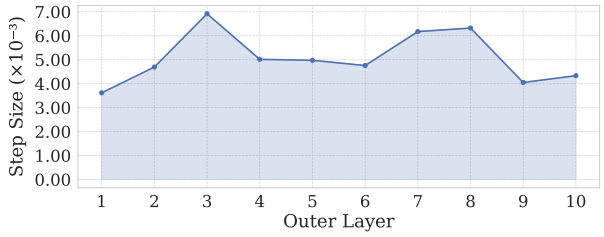
The penalty weights $\lambda_{\text{p}}^{(t)}$ and $\lambda_{\text{s}}^{(t)}$ in Fig. 3a do not follow a monotonic trend; instead, they form an alternating schedule. Rather than enforcing both constraints equally at every layer, the network learns to alternate their emphasis, which helps avoid stagnation in the non-convex landscape. The learned policy can be summarized in four phases:

- **Phase I — Baseline Exploration** (layers 1–3): Both penalties remain close to their static initializations ($\lambda_{\text{p}}^{(t)} \approx 10^5$, $\lambda_{\text{s}}^{(t)} \approx 5\times10^4$) to establish a stable initial trajectory.
- **Phase II — Power Priority** (layers 4–6): The network breaks the symmetry, sharply doubling the power penalty ($\lambda_{\text{p}}^{(t)} > 2 \times 10^5$) while relaxing the SINR penalty ($\lambda_{\text{s}}^{(t)} \approx 2.5\times10^4$). During this phase, the solver forces the beamformers to strictly respect the physical sum-power limit.
- **Phase III — SINR Priority** (layer 7): The logic inverts and the SINR penalty reaches its highest levels, while the power penalty drops to allow rapid recovery of user-QoS violations introduced during Phase II.
- **Phase IV — Terminal Power Enforcement** (layers 8–10): After the SINR peak, the network hands the trajectory back to the power barrier. By layer 10, the SINR penalty reaches its minimum value while the power penalty spikes, indicating that the final step effectively switches off additional SINR tightening and focuses on guaranteeing the absolute transmit-power limit at the output.

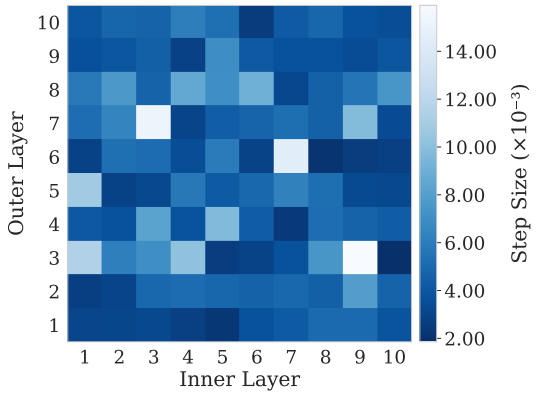
Figure 3c shows a matching oscillatory schedule for the inner step sizes $\eta^{(t,l)}$. Instead of decaying monotonically, the step size produces kinetic "bursts" at layers 3, 7, and 9, precisely when the penalty priorities change. This helps the solver move quickly between the power-feasible and SINR-feasible manifolds.



(a) Penalty Weights



(b) Outer Step Size



(c) Inner Step Size

Fig. 3: Layer-wise evolution of the learnable parameters of DU-SCA: (a) penalty weights $\lambda_p^{(t)}, \lambda_s^{(t)}$; (b) outer step sizes $\beta^{(t)}$; (c) inner step sizes $\eta^{(t,l)}$. Static classical-SCA values from Table II are overlaid for reference.

Physical and Algorithmic Interpretation. Together, these schedules define an optimization trajectory adapted to the non-convex geometry of the ISAC problem. Their roles are summarized below.

First, the learned outer step dictates the spatial covariance evolution through the sigmoid-gated update in the DU outer loop. Large early updates allow the solver to aggressively reshape the transmit covariance $\mathbf{R}_X$ to broaden the spatial illumination (rapidly driving down the CRB). In later layers, the sharp decay of the effective gate $\sigma(\beta^{(t)})$ transitions the solver into a fine-tuning regime, ensuring the beamformers stably settle onto the constraint boundaries without oscillating.

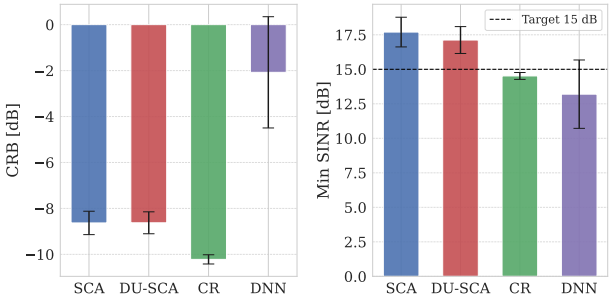
Second, the inner step size $\eta^{(t,l)}$ act as a *landscape adaptation mechanism*. By injecting kinetic "bursts" (e.g., at layers 3, 7 and 9) precisely when the penalty priorities invert, the network provides the inner solver with the necessary algorithmic momentum. This allows the trajectory to rapidly traverse the newly deformed surrogate landscape rather than stagnating when the feasibility cone abruptly shifts.

Finally, the alternating penalty schedule behaves like an autonomous alternating constraint projection. In congested ISAC scenarios, the power and SINR constraints create narrow, ill-conditioned valleys. By emphasizing them sequentially, the network avoids local minima that often trap static solvers. This coordinated policy helps DU-SCA synthesize low-CRB precoders under strict computational budgets.

C. Performance Comparison: Nominal Operating Point

At the nominal operating point, we evaluate the four solvers across three complementary dimensions to establish their baseline capabilities. First, we assess aggregate constraint satisfaction and average sensing accuracy to quantify overall efficacy. Second, we analyze the empirical distribution of these metrics to measure the statistical reliability of each solver across diverse channel realizations. Finally, we examine the per-iteration convergence dynamics and the resulting spatial beampatterns to understand the physical trade-offs adopted by the finite-budget solvers.

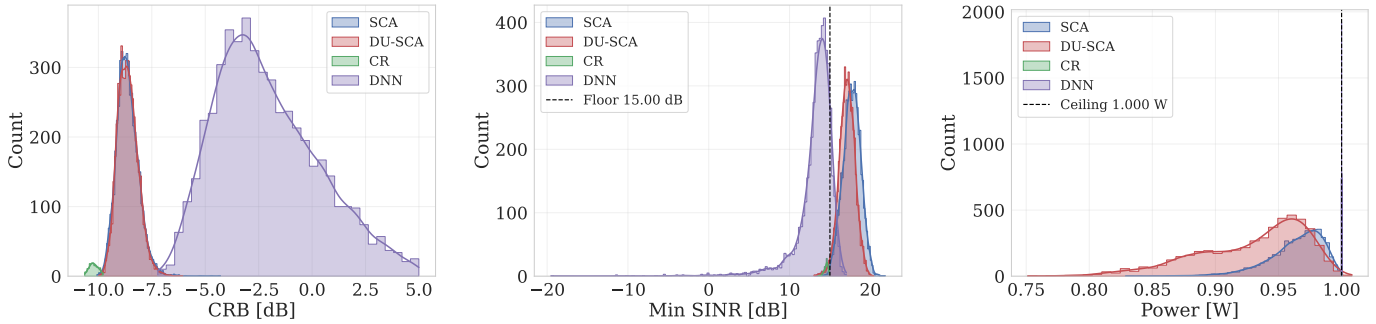
1) Aggregate Performance Summary: The bar charts in Fig. 4 summarize the aggregate CRB and minimum-SINR performance of the four benchmarks at the nominal operating point. The CR baseline attains the lowest average CRB of -10.22 dB, confirming its role as a quality-bound reference for covariance-level sensing optimization. This sensing bound is, however, obtained while essentially saturating the full power budget (≈ 1.0 W) and by averaging only 14.52 dB of minimum SINR, which translates into an 86.0% satisfaction rate and therefore a non-trivial outage probability relative to the $\Gamma = 15$ dB target.



(a) CRB (dB) (b) Min. SINR (dB)

Fig. 4: Bar charts summarising the nominal-operating-point averages (mean ± std) for all four solvers: (a) CRB (dB), (b) minimum SINR (dB) with the $\Gamma = 15$ dB target. Power behavior is discussed explicitly in the text.

The tuned classical SCA and the proposed DU-SCA are statistically indistinguishable in average sensing performance (-8.63 dB for both, with comparable standard deviations). The tuned classical baseline retains a modest edge in average minimum SINR (17.70 dB versus 17.12 dB) and satisfaction rate (99.94% versus 99.44%), while the proposed DU-SCA achieves this identical average CRB with lower mean transmit power (0.929 W versus 0.964 W). For power usage, the contrast is clear: both the CR baseline and the plain DNN saturate essentially the full 1.0 W budget, whereas the two proximal-SCA-based solvers operate below that boundary on average.



(a) CRB Density (b) SINR Density (c) Power Density

Fig. 5: Empirical distributions of per-realisation metrics at the nominal operating point for all four solvers: (a) CRB, (b) minimum SINR with $\Gamma = 15$ dB target, (c) transmit power with $P_T = 1$ W budget.

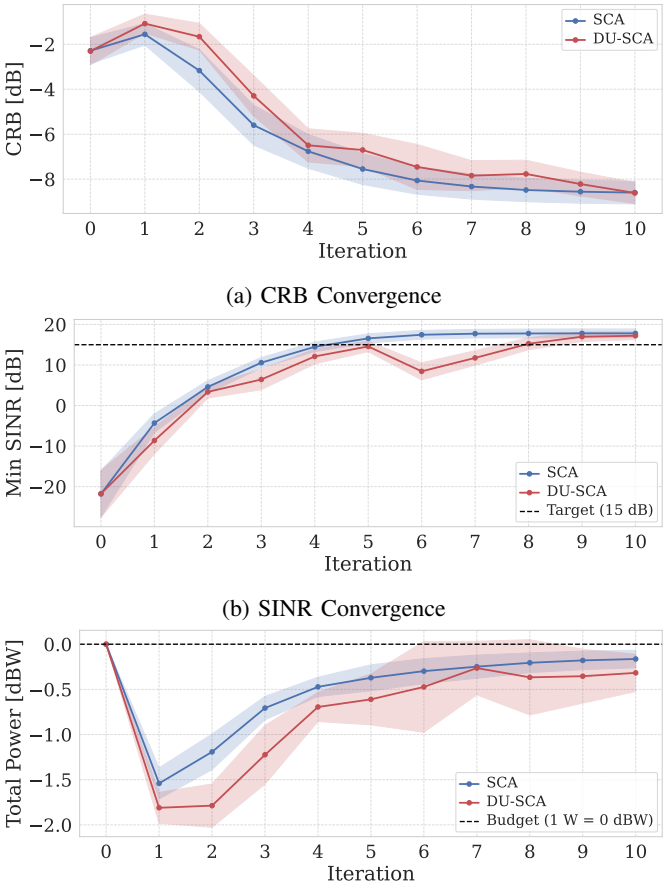
The plain DNN makes the cost of discarding the model-driven inductive bias explicit. Despite using the same unsupervised loss and training data as DU-SCA, it achieves only -2.08 dB average CRB, 13.20 dB average minimum SINR, and a poor 34.16% satisfaction rate while also saturating the 1.0 W power budget. Although it is the fastest solver (0.0084 ms), its performance is dominated by all model-based methods. This confirms that generic black-box regressors do not reliably synthesize valid ISAC precoders.

Both the classical SCA and DU-SCA exhibit tight error margins and reliable feasibility, demonstrating that the proposed unfolding architecture successfully preserves the mathematical rigor of its classical counterpart. Finally, the extreme variance and poor averages of the DNN visually confirm that replacing the iterative SCA geometry with a generic black-box network leads to systemic optimization failure at this operating point.

2) Statistical Reliability and Constraint Satisfaction: To investigate the reliability of the proposed method, we analyze the empirical distribution of the performance metrics over the test set. The SINR histogram in Fig. 5b shows that the tuned classical SCA baseline and the proposed DU-SCA concentrate almost all of their mass above the 15 dB target, consistent with the 99.94% and 99.44% satisfaction rates reported in the nominal-operating-point summary above. By contrast, the CR reference places a significant tail below the target (with mean 14.52 dB), since the Gaussian-randomization step enforces only the relaxed per-user constraint in expectation rather than the original non-convex constraint. The DNN distribution is visibly broader and centred well below the target, reflecting the 34.16% satisfaction level.

The power distribution in Fig. 5c complements this picture. The CR baseline and the DNN both saturate the power constraint almost deterministically, which for CR follows directly from the Karush-Kuhn-Tucker (KKT) conditions of the relaxed SDP [10] and for the DNN is imposed by the explicit Frobenius-norm projection in the output head. By contrast, the tuned classical SCA baseline and the proposed DU-SCA both operate below the 1 W boundary on average, with broadly similar power distributions. Although the classical solver exhibits a slightly higher mean transmit power, this gap is modest and should not be over-interpreted as a decisive source of the performance differences. The more relevant conclusion is simply that the two proximal-SCA-based methods attain comparable sensing quality without systematically saturating the available power budget, unlike the CR and DNN references.

3) Per-Iteration Convergence Trajectories: We now examine the per-iteration behaviour of the SCA and DU-SCA iterative solvers in Fig. 6.



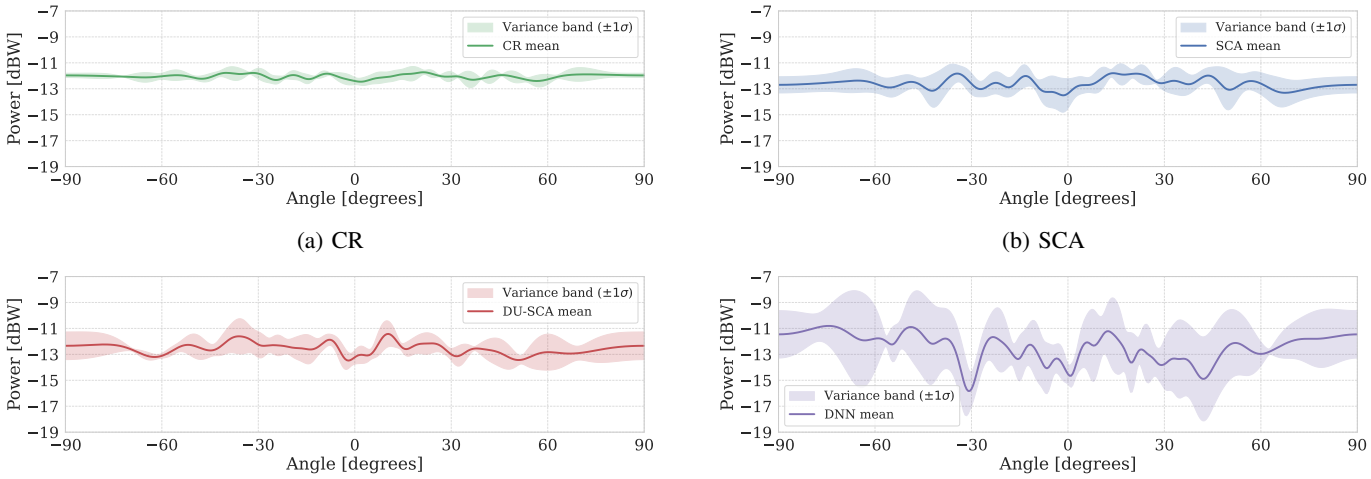
(a) CRB Convergence

(b) SINR Convergence

(c) Power Convergence

Fig. 6: Convergence trajectories of the iterative solvers at the nominal operating point: (a) CRB, (b) minimum SINR, and (c) transmit power. Shaded regions indicate 95% confidence intervals.

Both methods rapidly improve from a highly infeasible initialisation, but the tuned classical SCA baseline raises the



(a) CR

(b) SCA

(c) DU-SCA

(d) DNN

Fig. 7: Mean transmit beampatterns at the nominal operating point, with shaded $\pm 1\sigma$ bands across the sampled ensemble.

minimum SINR more quickly during the first half of the trajectory (Fig. 6a). In the convergence trace, the classical solver reaches 14.49 dB by iteration 4 and 16.56 dB by iteration 5, whereas the unfolded model reaches 12.11 dB and 14.57 dB at the same points before recovering in the later layers.

The sensing trajectories are also very close (Fig. 6b). By the final iteration, the two methods converge to nearly identical average CRB values, fully consistent with the nominal-operating-point average of approximately -8.63 dB reported above. Accordingly, the main conclusion from Fig. 6 is not uniform per-iteration dominance, but rather that the unfolded architecture reproduces a competitive optimization trajectory within a predetermined fixed-depth computation graph.

Consistent with the final statistics, the tuned classical SCA baseline ends the run at a slightly higher average transmit power, whereas the proposed DU-SCA attains a nearly identical sensing objective at a lower mean power level (Fig. 6c). This behavior indicates that the learned schedule primarily reshapes the power-allocation trajectory instead of simply enforcing a more aggressive approach to the power boundary at every iteration.

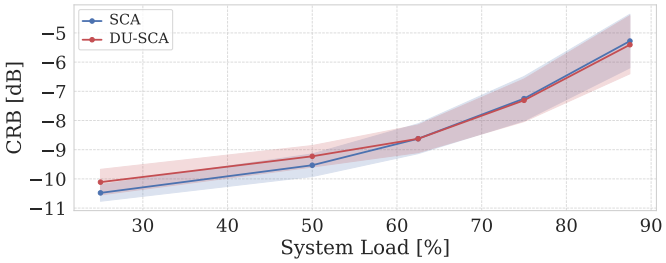
4) Spatial Interpretation via Beampatterns: While the nominal-point CRB and SINR summaries provide performance averages, the ensemble-mean beampatterns in Fig. 7 offer a deeper spatial interpretation of the precoder's behavior. These visualizations utilize $\pm 1\sigma$ variance bands to summarize the solver's consistency across the sampled ensemble, avoiding the risk of over-interpreting any single, potentially non-representative channel draw. Under the adopted non-parametric model, $\mathbf{H}_S$ is treated as an arbitrary unknown deterministic matrix, which corresponds to an extended-target scenario rather than a point target. The CRB in (4) therefore reduces to $(\sigma_S^2 N_r/L)\cdot \mathrm{tr}(\mathbf{R}_X^{-1})$ and is minimized only when the transmit covariance $\mathbf{R}_X = \mathbf{W}\mathbf{W}^H$ is well conditioned and spreads its eigen-energy across the angular domain; any concentration of radiated power into a narrow lobe inflates $\mathrm{tr}(\mathbf{R}_X^{-1})$ and degrades sensing quality. This analysis identifies whether the solvers maintain the broad spatial illumination required to prevent angular power collapse, which would otherwise inflate $\mathrm{tr}(\mathbf{R}_X^{-1})$ and severely degrade the CRB.

As illustrated by the spatial distributions in Fig. 7, the CR benchmark (Fig. 7a) remains the flattest and most isotropic reference, with an average across-angle standard deviation of only 0.294 dB and an average 3 dB beamwidth of $180°$. The tuned classical SCA and the proposed DU-SCA stay close to this extended-target behavior: as shown in Fig. 7b) and 7c), their average beamwidths are $180°$ and $167.5°$, while their mean across-angle dispersions are 0.778 dB and 0.747 dB, respectively. Importantly, the mean curves show that these two iterative solvers are not separated by a large systematic power offset: their mean pattern levels are nearly identical (-12.61 dBW for SCA versus -12.55 dBW for DU-SCA), while DU-SCA exhibits a slightly tighter worst-case variance band, with a peak standard deviation of 1.39 dB compared to 1.57 dB for the classical baseline. Accordingly, the main distinction revealed by Fig. 7 is not a difference in total radiated power, but that DU-SCA reproduces the broad covariance-shaping behavior of the tuned classical solver with a slightly more stable ensemble spread.

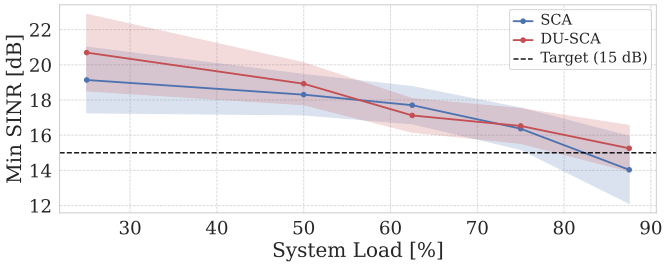
The mean-variance view also makes the DNN failure mode more explicit. Its average 3 dB beamwidth drops to $111.4°$, far below the model-based solutions, while its mean across-angle standard deviation rises to 1.98 dB and its local variability reaches 3.69 dB at the most unstable angles. The resulting variance envelope spans from -18.13 to -8.09 dBW, which is dramatically broader than the model-based bands and indicates strong sample-to-sample lobe migration rather than stable wide-angle illumination. This ensemble instability is the covariance signature of a black-box solution that keeps rediscovering narrow, point-target-like beams instead of the well-conditioned extended-target covariances favored by the non-parametric CRB. Hence, the new figures strengthen the original conclusion: the model-driven inductive bias is not merely improving average metrics, it is enforcing the correct spatial geometry consistently across channel realizations.

D. Scalability and Loading Analysis

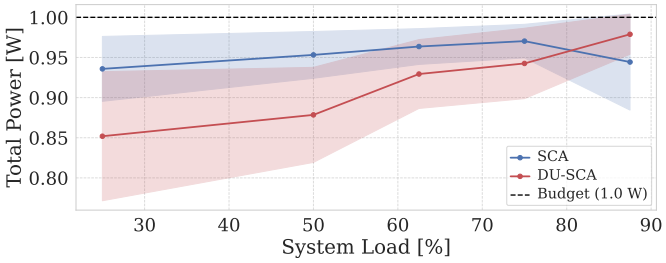
To evaluate the impact of spatial loading, we compare the tuned classical SCA and the proposed DU-SCA under varying user demands. As illustrated in Fig. 8, we sweep the spatial load factor K/N_t from a lightly loaded $K = 4$ (25%) to a heavily congested $K = 14$ (87.5%). Analyzing these performance trajectories with 95% confidence intervals reveals three distinct operational regimes.



(a) CRB vs. User loading



(b) Min. SINR (dB) vs. User Loading



(c) Power (W) vs. User Loading

Fig. 8: Performance sensitivity to user loading K: (a) CRB, (b) minimum SINR, and (c) transmit power. Shaded regions indicate 95% confidence intervals.

In the low-load regime ($K = 4$ and $K = 8$), the spatial degrees of freedom are abundant, and both methods easily maintain feasibility. Under these relaxed conditions, the tuned classical SCA achieves slightly lower CRB values, whereas the proposed DU-SCA prioritizes communication robustness, delivering higher SINR margins and utilizing lower average power. The nominal operating point ($K = 10$) serves as the crossover threshold: the two solvers become statistically tied in sensing performance (-8.626 dB versus -8.624 dB), with the classical baseline retaining a modest advantage in average minimum SINR (17.71 dB versus 17.12 dB).

The critical advantage of the unfolded architecture emerges strictly as the system enters the overloaded regime ($K \geq 12$). As the non-convex constraints become increasingly difficult to satisfy, the classical SCA baseline begins to degrade, ultimately failing to maintain the 15 dB communication target at $K = 14$ (averaging only 14.03 dB). By contrast, by making better usage of the power, DU-SCA successfully adapts its learned trajectory to preserve average feasibility (15.26 dB) while also edging out the classical method in sensing quality (-5.40 dB versus -5.28 dB). This confirms that while the classical method excels in lightly constrained environments, the DU-SCA architecture provides superior robustness and reliability in highly congested ISAC networks.

E. Tradeoff Analysis

To characterize the fundamental trade-off between sensing capability and communication reliability, we trace the tradeoff frontier of the system by sweeping the target SINR constraint Γ from 5 dB to 25 dB. Fig. 9 summarizes the resulting minimal CRB and constraint satisfaction margins.

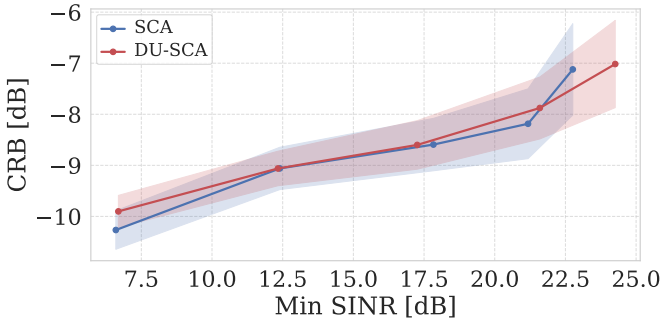


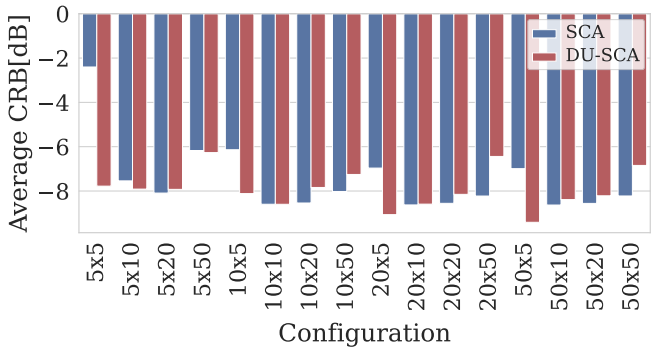
Fig. 9: Sensing-communication trade-off frontier (average CRB versus minimum SINR).

The resulting Pareto frontiers do not show strict dominance; instead, they reveal an operating-point-dependent trade-off between communication margin and sensing optimality. We observe two distinct operational regimes:

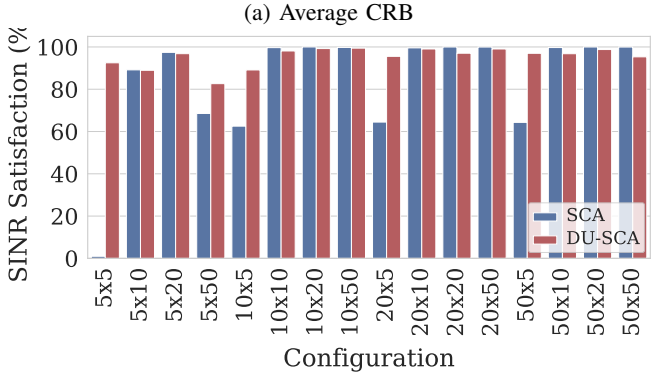
1) Relaxed Constraints: Under loose constraints ($\Gamma \leq 10$ dB), both methods successfully locate valid solutions and remain very close in terms of sensing performance. At $\Gamma = 5$ dB, the tuned classical SCA baseline attains a slightly lower CRB (-10.27 dB versus -9.90 dB), whereas at $\Gamma = 10$ dB (LCM: Acho que aqui deveria ser $\Gamma = 12.5$ dB) the two methods are essentially tied (-9.06 dB for both) with only a marginal difference in achieved SINR.

2) Stringent Constraints: As communication requirements become stringent ($\Gamma \geq 15$ dB), the trade-off shifts. At $\Gamma = 15$ dB, both methods remain comfortably feasible, the tuned classical SCA baseline attains a higher achieved SINR (17.83 dB versus 17.26 dB), and the average CRB values remain almost identical. At $\Gamma = 20$ dB and $\Gamma = 25$ dB, the proposed DU-SCA provides a larger communication margin, reaching 21.59 dB and 24.26 dB, respectively, while the tuned classical SCA baseline attains 21.18 dB and 22.76 dB. This communication benefit is accompanied by a modest sensing penalty, since the tuned classical SCA baseline yields slightly lower CRB values at these stringent targets. Hence, the results support the interpretation that the unfolded architecture becomes more communication-oriented as the target tightens, rather than uniformly improving both objectives simultaneously.

3) Computational Complexity and Deployment Feasibility: To evaluate practical deployment feasibility, we analyze the computational cost of the solvers from two complementary perspectives: iteration-budget efficiency and absolute inference latency. First, we establish the algorithmic convergence by sweeping the network depth $T \in \{5, 10, 20\}$ and the number of inner steps $L \in \{5, 10, 20, 30, 50\}$ against the total gradient evaluations ($T \times L$).



(a) Average CRB



(b) Constraint Satisfaction

Fig. 10: Performance under a strict total iteration budget $T \times L$ for the tuned classical SCA and DU-SCA: (a) average CRB, (b) joint constraint-satisfaction rate versus total gradient evaluations.

Figure 10 shows that the main advantage of DU-SCA appears in the ultra-low-budget regime. With only 25 gradient evaluations (5×5), the unfolded model already reaches 97.7% SINR satisfaction and a CRB of -7.91 dB, whereas the tuned classical SCA baseline remains far from convergence, achieving only 4.86% satisfaction and -2.51 dB.

This gap narrows as the budget increases. By 50 iterations the two methods are already close in feasibility, and by 100 iterations the tuned classical baseline slightly exceeds DU-SCA in satisfaction rate while the two remain tied in average CRB. The main benefit of DU-SCA is therefore faster convergence at limited depth, not a different asymptotic operating point.

Table III highlights how iteration counts translate to runtime stability. While classical SCA converges earlier at relaxed targets (5 dB), it progressively exhausts its 100-iteration budget as constraints tighten ($\Gamma \geq 20$ dB). Crucially, its high standard deviations reveal significant computational jitter induced by the Armijo line search. In contrast, DU-SCA executes a fixed $T \times L = 100$ gradient evaluations per realization. This entirely eliminates runtime jitter, guaranteeing the deterministic execution required for real-time ISAC framing.

TABLE III: Effective iteration count of the tuned classical SCA vs. DU-SCA under a 100-iteration budget at $N_t = 16$, $K = 10$.

Target Γ	Method	Mean	Std	Min–Max
5 dB	SCA	74.3	14.98	50–100
5 dB	**DU-SCA**	**100.0**	**0.0**	**100–100**
10 dB	SCA	81.5	14.37	50–100
10 dB	**DU-SCA**	**100.0**	**0.0**	**100–100**
15 dB	SCA	91.1	10.92	60–100
15 dB	**DU-SCA**	**100.0**	**0.0**	**100–100**
20 dB	SCA	97.7	5.79	70–100
20 dB	**DU-SCA**	**100.0**	**0.0**	**100–100**
25 dB	SCA	98.2	4.73	70–100
25 dB	**DU-SCA**	**100.0**	**0.0**	**100–100**

Crucially, the iteration efficiency and deterministic execution of DU-SCA translate directly into the second evaluation metric: a persistent absolute latency advantage. To quantify this, we measure the per-sample wall-clock inference latency for all four solvers on a common evaluation platform at the nominal operating point. As visualized on the logarithmic axis of Fig. 11 and detailed in Table IV, the timing distributions reveal a massive five-order-of-magnitude spread between the theoretical quality reference and the deployable methods.

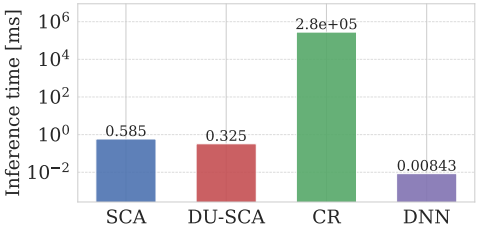


Fig. 11: Per-sample inference latency at the nominal operating point ($N_t = 16$, $K = 10$, $\Gamma = 15$ dB, 5000 channels) for the four solvers, logarithmic axis.

TABLE IV: Measured per-sample inference latency at the nominal operating point ($N_t = 16$, $K = 10$, $\Gamma = 15$ dB, 5000 Monte Carlo channels).

Method	Latency (ms)	Relative to DU-SCA
CR [5], [10]	2.80×10^5	$\sim 8.6 \times 10^5\times$
SCA (tuned)	0.585	1.80×
DU-SCA	**0.325**	**1.00×**
DNN	0.0084	0.026×

The CR benchmark requires an unviable 2.80×10^5 ms per sample, rendering it completely incompatible with low-latency ISAC deadlines. Conversely, while the plain DNN executes in just 0.0084 ms, this isolated speed comes at cost of feasibility failures demonstrated earlier. Focusing strictly on the deployable solvers, the proposed DU-SCA attains 0.325 ms per sample against the 0.585 ms of the tuned classical SCA baseline. This corresponds to an average speedup of approximately 1.80×. By successfully co-optimizing sensing accuracy, strict constraint satisfaction, and real-time execution

speed, these results confirm that DU-SCA uniquely defines the Pareto frontier for this architecture.

4) Inference Time Scaling vs. Unrolling Depth: To further characterize the deployment feasibility of the proposed framework, we profiled the absolute per-sample inference latency as a function of the inner (L) and outer (T) unrolling depths. As illustrated in Fig. 12, hardware profiling demonstrates that execution time scales strictly linearly with the total number of gradient evaluations ($T \times L$), providing absolute predictability for real-time scheduling.

For instance, an aggressively truncated 5×5 configuration (25 total iterations) requires approximately 0.32 ms per sample. Doubling the computational budget to 50 total iterations—whether structured as an outer-heavy 10×5 configuration (0.63 ms) or an inner-heavy 5×10 configuration (0.59 ms)—proportionally scales the latency. At the nominal 10×10 configuration (100 total iterations), the measured inference time is 1.15 ms. While this requires a slightly larger time budget than the heavily truncated setups, it guarantees high-fidelity convergence while maintaining strict deterministic bounds.

Conversely, extremely deep configurations, such as 20×50 (1000 total iterations), demand 10.69 ms per sample, pushing the execution outside the boundaries of practical real-time ISAC coherence windows. Ultimately, this strictly linear and jitter-free scaling empowers system designers to precisely calibrate the unrolling depth, explicitly trading a fraction of asymptotic sensing performance for guaranteed compliance with sub-millisecond hardware deadlines.

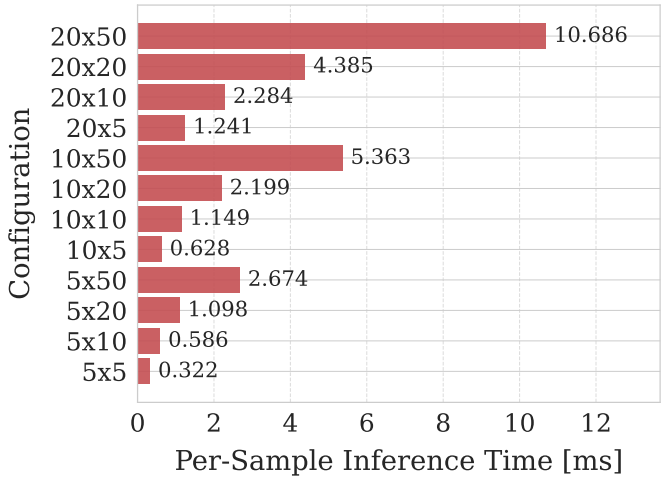


Fig. 12: Per-sample inference time scaling as a function of the total unfolded iterations ($T \times L$).

5) Training Time and Computational Overhead: The offline training phase is heavier per epoch than the black-box baseline because the unfolded network explicitly evaluates the proximal-SCA operations inside the computational graph. On the profiled 2048-sample setup, DU-SCA requires about 9.59 seconds per epoch, whereas the plain DNN requires 0.187 seconds.

Even so, the total offline cost remains modest because DU-SCA learns only $T(L+5) = 150$ scalar hyperparameters. In practice, it converges in fewer than 100 epochs, corresponding to roughly 15 minutes on standard desktop hardware, which keeps retraining feasible when the system dimensions change.

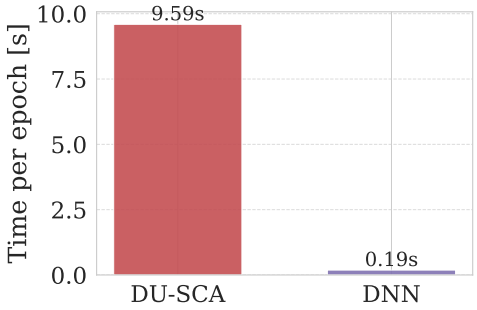


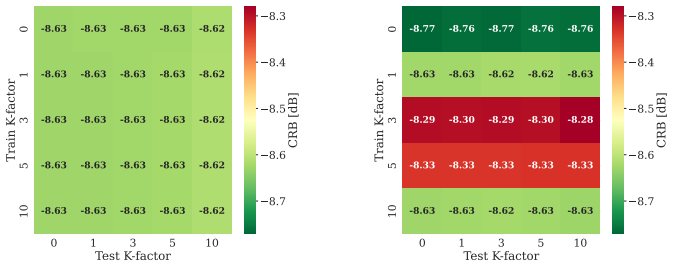
Fig. 13: Offline computational overhead comparison per training epoch.

F. Generalization Discussion

The generalization behavior of DU-SCA differs fundamentally from that of black-box DNNs. Rather than learning the full channel-to-precoder mapping, DU-SCA learns only $T(L+5) = 150$ optimizer hyperparameters, while the precoder itself is still produced by the embedded physics-based SCA iterations during inference.

This leads to two structural properties. First, the forward pass remains a valid SCA-type solver for any channel realization, so distribution shift does not produce arbitrary or physically invalid outputs. Second, when the learned schedules are mismatched to a new distribution, performance degrades toward that of a fixed-hyperparameter SCA rather than collapsing abruptly.

If the system dimensions change, retraining is still recommended because the optimization structure itself changes. Even so, the cost remains modest because the model contains only 150 learnable scalars.



(a) Tuned classical SCA (b) Proposed DU-SCA

Fig. 14: Cross-channel robustness heatmaps of average CRB (dB) under Rician factor mismatch κ: (a) tuned classical SCA, and (b) DU-SCA.

To empirically validate these structural guarantees, we evaluate the solvers under severe channel distribution shifts. Specifically, we induce a propagation mismatch by varying the Rician fading factor $\kappa \in \{0, 1, 3, 5, 10\}$, where $\kappa = 0$ represents the standard Rayleigh fading used during the initial offline training, and higher values introduce increasingly dominant Line of Sight (LOS)) components. The cross-channel heatmaps in Fig. 14 capture this mismatch, plotting the average CRB when a solver optimized for a specific κ_{train} (vertical

axis) is evaluated across different test distributions κ_{test} (horizontal axis).

Examining training sensitivity (vertical axis) provides a clearer picture of the learned optimizer's behavior. For the tuned classical SCA, the operating point is nearly invariant across $\kappa_{\text{train}} \in \{0, 1, 3, 5, 10\}$, a direct consequence of deploying a universally robust static parameter set. For DU-SCA, the schedules learned during offline training are highly specialized to their respective training distributions. While a schedule trained on pure Rayleigh data ($\kappa_{\text{train}} = 0$) achieves an excellent -8.77 dB, training on intermediate mixed-fading regimes (e.g., $\kappa_{\text{train}} = 3$) yields a slightly relaxed CRB of -8.29 dB. The key takeaway, however, is not strict optimality across all shifts, but structural resilience: even when the learned scheduler is no longer perfectly optimal for a different data distribution, the method does not fail catastrophically. Crucially, across all tested distribution shifts, DU-SCA successfully maintained 100% constraint satisfaction. Ultimately, while DU-SCA achieves peak performance when trained on matched channel statistics, its underlying proximal-SCA geometry ensures safe, continuous, and fully feasible system operation even if the real-world propagation environment diverges.

VII. Conclusion

This work investigated the trade-off between computational complexity and strict latency requirements in real-time ISAC beamforming. To address this challenge, we introduced DU-SCA, a model-driven architecture that unfolds the full nested proximal SCA solver into a fixed-depth computational graph. By jointly learning layer-wise optimization schedules and eliminating the Armijo backtracking stage, the online inference pipeline is made entirely deterministic.

Numerical results demonstrated that DU-SCA performs on par with a finely tuned classical SCA at nominal operating points, while significantly outperforming a plain black-box DNN and avoiding the prohibitive latency of SDR-based bounds. The primary advantages of DU-SCA emerge under constrained computational budgets and overloaded spatial scenarios, where it effectively maintains physical feasibility while the classical baseline collapses. Furthermore, DU-SCA achieves a lower overall radiated-pattern level, delivers a consistent wall-clock speedup (averaging $1.8\times$), and exhibits graceful degradation under moderate channel distribution shifts.

Regarding scalability, the ultra-compact parameterization of DU-SCA ensures that offline retraining overhead remains negligible when system dimensions change, circumventing the combinatorial explosion inherent to manual grid-search tuning. Overall, the proposed DU-SCA framework establishes a practical foundation for real-time, interpretable beamforming in 6G ISAC systems. Future research will extend this paradigm toward robust precoding under Channel State Information (CSI) uncertainty and wideband millimeter-wave deployments.

References

[1] F. Liu, C. Masouros, A. P. Petropulu, H. Griffiths, and L. Hanzo, "Joint radar and communication design: Applications, state-of-the-art, and the road ahead," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3834–3862, 2020.

[2] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1728–1767, 2022.

[3] A. Liu, Z. Huang, M. Li, Y. Wan, W. Li, T. X. Han, C. Liu, R. Du, D. K. P. Tan, J. Lu, Y. Shen, F. Colone, and K. Chetty, "A survey on fundamental limits of integrated sensing and communication," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 994–1034, 2022.

[4] B. Paul, A. R. Chiriyath, and D. W. Bliss, "Survey of RF communications and sensing convergence research," *IEEE Access*, vol. 5, pp. 252–270, 2017.

[5] F. Liu, L. Zhou, C. Masouros, A. Li, W. Luo, and A. Petropulu, "Toward dual-functional radar-communication systems: Optimal waveform design," *IEEE Transactions on Signal Processing*, vol. 66, no. 16, pp. 4264–4279, 2018.

[6] Z. Cheng, B. Liao, S. Shi, Z. He, and J. Li, "Co-design for overlaid MIMO radar and downlink MISO communication systems via Cramér-Rao bound minimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 24, pp. 6227–6240, 2019.

[7] F. Liu, C. Masouros, A. Li, and T. Ratnarajah, "Robust MIMO beamforming for cellular and radar coexistence," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 374–377, 2017.

[8] H. Hua, J. Xu, and T. X. Han, "Transmit beamforming optimization for integrated sensing and communication," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[9] ——, "Optimal transmit beamforming for integrated sensing and communication," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 10 588–10 603, 2023.

[10] F. Liu, Y.-F. Liu, A. Li, C. Masouros, and Y. C. Eldar, "Cramér-Rao bound optimization for joint radar-communication beamforming," *IEEE Transactions on Signal Processing*, vol. 70, pp. 240–253, 2022.

[11] F. Liu, Y.-F. Liu, C. Masouros, A. Li, and Y. C. Eldar, "A joint radar-communication precoding design based on Cramér-Rao bound optimization," in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 1–6.

[12] Z. He, W. Xu, H. Shen, D. W. K. Ng, Y. C. Eldar, and X. You, "Full-duplex communication for ISAC: Joint beamforming and power optimization," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 9, pp. 2920–2936, 2023.

[13] N. T. Nguyen, N. Shlezinger, Y. C. Eldar, and M. Juntti, "Multiuser MIMO wideband joint communications and sensing system with subcarrier allocation," *IEEE Transactions on Signal Processing*, vol. 71, pp. 2997–3013, 2023.

[14] N. T. Nguyen, L. V. Nguyen, N. Shlezinger, Y. C. Eldar, A. L. Swindlehurst, and M. Juntti, "Joint communications and sensing hybrid beamforming design via deep unfolding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 5, pp. 901–916, 2024.

[15] M. Temiz, Y. Zhang, Y. Fu, C. Zhang, C. Meng, O. Kaplan, and C. Masouros, "Deep-learning-based techniques for integrated sensing and communication systems: State-of-the-art, challenges, and opportunities," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 5940–5968, 2025.

[16] J. Zhang, C. Masouros, F. Liu, Y. Huang, and A. L. Swindlehurst, "Low-complexity joint radar-communication beamforming: From optimization to deep unfolding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 19, no. 5, pp. 856–870, 2025.

[17] Q. Gao, R. Zhong, Y. Zou, H. Shin, and Y. Liu, "Deep learning-based beamforming optimization for ISAC systems: A low-complexity and transferable framework," *IEEE Transactions on Wireless Communications*, vol. 25, pp. 8754–8768, 2026.

[18] S. Deka, K. Deka, N. T. Nguyen, S. Sharma, V. Bhatia, and N. Rajatheva, "Comprehensive review of deep unfolding techniques for next-generation wireless communication systems," *arXiv preprint arXiv:2502.05952*, 2025. [Online]. Available: https://arxiv.org/abs/2502.05952

[19] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 77–83, 2019.

[20] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014. [Online]. Available: https://arxiv.org/abs/1409.2574

[21] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 399–406.

[22] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2019, pp. 266–271.

[23] J. Zhang, Y. Zhu, N. Zhao, S. Jin, X. Wang, D. W. K. Ng, and N. Al-Dhahir, "Deep unfolding learning aided ISAC transceiver design," *IEEE Transactions on Wireless Communications*, vol. 24, no. 10, pp. 8681–8696, 2025.

[24] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[25] K. M. Attiah and W. Yu, "Beamforming design for integrated sensing and communications using uplink-downlink duality," in *2024 IEEE International Symposium on Information Theory (ISIT)*, Athens, Greece, Jul. 2024.

[26] ——, "Uplink-downlink duality for beamforming in integrated sensing and communications," *arXiv preprint arXiv:2509.13661*, 2025.

[27] J.-G. Wu, B. Jiang, X.-X. Li, Y.-F. Liu, and J.-H. Yuan, "A new adaptive balanced augmented lagrangian method with application to ISAC beamforming design," *Journal of the Operations Research Society of China*, pp. 1–24, 2025.

[28] X. Zhao, M. Li, Y.-F. Liu, Q. Shi, and A. M.-C. So, "Optimal low-dimensional structures of ISAC beamforming: Theory and efficient algorithms," *arXiv preprint arXiv:2602.07502*, 2026.

[29] M. Zhu, L. Li, S. Xia, and T.-H. Chang, "Information and sensing beamforming optimization for multi-user multi-target MIMO ISAC systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 1, p. 15, 2023.

[30] T. Fang, N. T. Nguyen, and M. Juntti, "Low-complexity Cramér-Rao lower bound and sum rate optimization in ISAC systems," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[31] T. Fang, M. Ma, M. Juntti, N. Shlezinger, A. L. Swindlehurst, and N. T. Nguyen, "Optimal ISAC beamforming structure and efficient algorithms for sum rate and CRLB balancing," *arXiv preprint arXiv:2503.09489*, 2025.

[32] Y. Liu, Q. Hu, Y. Cai, G. Yu, and G. Y. Li, "Deep-unfolding beamforming for intelligent reflecting surface assisted full-duplex systems," *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 4784–4800, 2022.

[33] B. Li, G. Verma, and S. Segarra, "Graph-based algorithm unfolding for energy-aware power allocation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1359–1373, 2023.

[34] J. Zhang, M. Liu, J. Tang, N. Zhao, D. Niyato, and X. Wang, "Joint design for RIS-aided ISAC via deep unfolding learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 1, pp. 349–362, 2025.

[35] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[36] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511804441

[38] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, 2014.

[39] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[40] A. Liu, V. K. N. Lau, and B. Kananian, "Stochastic successive convex approximation for non-convex constrained stochastic optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4189–4203, 2019.

[41] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[42] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning (ICML)*, 2013, pp. 1139–1147.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2013, pp. 1310–1318.

[44] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[45] R. S. Sutton, "Adapting bias by gradient descent: an incremental version of delta-bar-delta," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, ser. AAAI'92. AAAI Press, 1992, p. 171–176.

[46] Q. Shi and M. Hong, "Penalty dual decomposition method for non-smooth nonconvex optimization—part I: Algorithms and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4108–4122, 2020.

[47] Q. Shi, M. Hong, X. Fu, and T.-H. Chang, "Penalty dual decomposition method for nonsmooth nonconvex optimization—part II: Applications," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4242–4257, 2020.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[49] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.

Rafael Marasca Martins received the B.S. degree in computer engineering from the Federal University of Technology – Paraná (UTFPR), Brazil, in 2024. He is currently pursuing the M.Sc. degree in electrical engineering at the State University of Londrina (UEL), Brazil. His professional experience includes hardware and software development for embedded systems. His current research interests include machine learning, telecommunications, and signal processing.

Bruno Felipe Costa received the B.S. and M.Sc. degrees in electrical engineering from the State University of Londrina (UEL), Brazil, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in electrical engineering through a split-site doctoral program between UEL and Aalborg University, Denmark. His research interests focus on Low Earth Orbit (LEO) satellite systems for Integrated Sensing and Communication (ISAC).

Luis Carlos Mathias received the B.S. degree in physics in 2007, the M.Sc. degree in electrical engineering in 2013, the B.S. degree in electrical engineering in 2018, and the Ph.D. degree in electrical engineering in 2019, all from the State University of Londrina (UEL), Brazil. He held postdoctoral research positions at UEL and the University of Manitoba (UM), Winnipeg, MB, Canada, in 2025. He is currently a Professor with the Federal University of Technology – Paraná (UTFPR), Toledo, Brazil. His research interests mainly include detection, estimation, and energy and spectral efficiency in wireless optical communication systems.

Taufik Abrão received the B.S., M.Sc., and Ph.D. degrees in electrical engineering from the Polytechnic School of the University of São Paulo, São Paulo, Brazil, in 1992, 1996, and 2001, respectively. He also completed two postdoctoral research fellowships. Since 1997, he has been with the Communications Group, Department of Electrical Engineering, State University of Londrina (UEL), Paraná, Brazil, where he is currently a Full Professor in telecommunications and the Head of the Telecomm and Signal Processing Lab. He is a Productivity Researcher with the CNPq Brazilian Agency (Pq-1C).