Explainable AI and Optimization Theory based Wireless Radio Resource Management

by

Nasir Khan

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

July 28, 2025

Explainable AI and Optimization Theory based Wireless Radio Resource Management

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Nasir Khan

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Sinem Çöleri

_____

Prof. Öznur Özkasap

_____

Prof. Ertuğrul Başar

_____

Prof. Özgür Gürbüz

_____

Prof. Ali Emre Pusane

Date: _____

To my parents, Ahmad Khan and Bashir-un-Nisa, whose prayers, guidance, and love continue to illuminate my path.

ABSTRACT

Explainable AI and Optimization Theory based Wireless Radio Resource Management

Nasir Khan

Doctor of Philosophy in Electrical and Electronics Engineering

July 28, 2025

The International Telecommunication Union's IMT-2030 framework identifies "Integrated artificial intelligence (AI) and Communication" as one of the core directions for 6G networks. This shift toward AI-driven solutions necessitates greater transparency in decision-making processes, where explainability is pivotal in establishing trust in AI systems, allowing network operators and engineers to understand, validate, and troubleshoot the decisions made by deep learning (DL) models. Additionally, robustness against out-of-distribution inputs and adversarial attacks is crucial to ensure reliable performance in diverse and evolving environments. These two factors—explainability and robustness—are essential to fulfilling the broader goals of AI-native 6G networks, where AI is not just an enhancement but a foundational component integrated into communication systems.

This thesis examines the application of Explainable AI (XAI) for robust and transparent radio resource management (RRM) in AI-empowered 6G networks. First, the significance of this research is presented by unveiling the merits of explainability and robustness for 6G RRM, outlining a range of core explainability and robustness techniques for efficient RRM. Then, the practical implementation of these XAI techniques is analyzed, including a quantitative evaluation of their impact on key performance indicators (KPIs) in specific 6G scenarios. Two case studies are presented, showcasing their application in model simplification and enhancing RRM robustness.

The first study proposes an explainable and robust DL-based framework for beam management in a millimeter-wave (mmWave) multiple-input multiple-output (MIMO) communications system. To minimize signaling overhead for beam alignment, a novel model-agnostic, feature relevance-oriented XAI framework is designed

to rank and prioritize input features, reducing input size and beam sweeping time. To incorporate transparency and resilience into the DL-based beam alignment engine (BAE), a defense mechanism against adversarial inputs is developed, examining internal representations learned by deep neural networks (DNN) to provide interpretability and robustness against malicious/out-of-distribution inputs. The second case study focuses on deep reinforcement learning (DRL)-based RRM in vehicular networks for joint transmit power and spectrum allocation. First, a model-agnostic XAI-based methodology is devised to explain the inference process of multiple trained DRL agents. Then, a novel XAI-based feature importance ranking algorithm using Shapley additive explanations (SHAP) is developed to simplify the DRL agents' input state size, followed by an automated feature selection strategy to reduce model complexity. By eliminating low-importance features, the methodology produces a simplified model with fewer network parameters and lower training time while maintaining reasonable performance.

This thesis further introduces a novel DRL-based framework for joint power control and block length allocation to minimize the worst-case decoding error probability for ultra-reliable and low-latency communication (URLLC)-based vehicular networks. Initially, an algorithm grounded in optimization theory is developed based on deriving the joint convexity of the decoding error probability in the block length and transmit power variables within the region of interest. Subsequently, inspired by event-triggered control systems, an efficient event-triggered DRL-based algorithm is proposed to solve the joint optimization problem. Incorporating event-triggered learning into the DRL framework enables assessing whether to initiate the DRL process, thereby reducing the number of DRL process executions while maintaining reasonable reliability performance. The results demonstrate a noteworthy reduction in training and runtime complexity compared to the conventional optimization-based solutions.

ÖZETÇE

Doktora Tez Başlığı
Nasir Khan
Elektrik ve Elektronik Mühendisliği, Doktora
28 Temmuz 2025

Uluslararası Telekomünikasyon Birliği'nin IMT-2030 çerçevesi, "Entegre yapay zeka (AI) ve İletişim"i 6G ağlarının temel yönlerinden biri olarak tanımlıyor. Yapay zeka odaklı çözümlere yönelik bu geçiş, karar verme süreçlerinde daha fazla şeffaflık gerektiriyor; burada açıklanabilirlik, yapay zeka sistemlerine güven oluşturmada çok önemli, ağ operatörlerinin ve mühendislerin derin öğrenme (DL) modelleri tarafından alınan kararları anlamalarına, doğrulamalarına ve sorunlarını gidermelerine olanak tanıyor. Ayrıca, dağıtım dışı girdilere ve düşmanca saldırılara karşı dayanıklılık, çeşitli ve gelişen ortamlarda güvenilir performans sağlamak için çok önemlidir. Bu iki faktör (açıklanabilirlik ve sağlamlık), yapay zekanın yalnızca bir geliştirme değil aynı zamanda iletişim sistemlerine entegre edilmiş temel bir bileşen olduğu yapay zekaya özgü 6G ağlarının daha geniş hedeflerini gerçekleştirmek için gereklidir.

Bu tez, Yapay Zeka destekli 6G ağlarında sağlam ve şeffaf radyo kaynağı yönetimi (RRM) için Açıklanabilir Yapay Zeka (XAI) uygulamasını incelemektedir. İlk olarak, bu araştırmanın önemi, 6G RRM için açıklanabilirlik ve sağlamlığın yararları verilerek, bir dizi temel açıklanabilirlik ve sağlamlığın ana hatları üzerinden sunulmaktadır. Daha sonra, bu XAI tekniklerinin pratik uygulaması, belirli 6G senaryolarında temel performans göstergeleri (KPI'ler) üzerindeki etkilerinin niceliksel bir değerlendirmesiyle analiz ediliyor. Model basitleştirme ve RRM sağlamlığının arttırılmasındaki uygulamalarını gösteren iki vaka çalışması sunulmaktadır.

İlk vaka çalışması, milimetre dalga (mmWave) çoklu giriş çoklu çıkış (MIMO) iletişim sisteminde ışın yönetimi için açıklanabilir ve sağlam bir DL tabanlı çerçeve önermektedir. Işın hizalaması için sinyalleme yükünü en aza indirmek amacıyla, giriş özelliklerini sıralayıp öncelik sırasına koymak, giriş boyutunu ve ışın tarama süresini azaltmak için yeni bir model-agnostik, özellik alaka odaklı XAI çerçevesi tasarlanmıştır. Şeffaflığı ve dayanıklılığı DL tabanlı ışın hizalama motoruna (BAE) dahil

etmek için, kötü niyetli/dağıtım dışı girdilere karşı yorumlanabilirlik ve sağlamlık sağlamak için derin sinir ağları (DNN) tarafından öğrenilen dahili temsilleri inceleyen, düşmanca girdilere karşı bir savunma mekanizması geliştirilmiştir. İkinci vaka çalışması, ortak iletim gücü ve spektrum tahsisi için araç ağlarında derin takviyeli öğrenme (DRL) tabanlı RRM'ye odaklanmaktadır. İlk olarak, birden fazla eğitilmiş DRL aracısının çıkarım sürecini açıklamak için model-agnostik XAI tabanlı bir metodoloji tasarlanmıştır. Daha sonra, DRL aracılarının giriş durumu boyutunu basitleştirmek için Shapley eklemeli açıklamaları (SHAP) kullanan yeni bir XAI tabanlı özellik önem sıralaması algoritması geliştirilir ve ardından model karmaşıklığını azaltmak için otomatik bir özellik seçimi stratejisi uygulanır. Düşük öneme sahip özellikleri ortadan kaldırarak, metodoloji performansını korurken daha az ağ parametresi ve daha düşük eğitim süresine sahip basitleştirilmiş bir model üretir.

Bu tez ayrıca, ultra güvenilir ve düşük gecikmeli iletişim (URLLC) tabanlı araç ağları için en kötü durum kod çözme hatası olasılığını en aza indirmek amacıyla ortak güç kontrolü ve blok uzunluğu tahsisi için yeni bir DRL tabanlı çerçeve sunmaktadır. Başlangıçta bloktaki kod çözme hatası olasılığının ilgilenilen bölgedeki uzunluk ve iletim gücü değişkenlerindeki ortak dışbükeyliğini kullanan optimizasyon teorisine dayalı bir algoritma geliştirilmiştir. Daha sonra, ortak optimizasyon problemini çözmek için olayla tetiklenen kontrol sistemlerinden ilham alınarak, etkin, olayla tetiklenen DRL tabanlı bir algoritma önerilmiştir. Birleştirme DRL çerçevesine olayla tetiklenen öğrenmeyi dahil etmek DRL sürecinin başlatılıp başlatılmayacağına karar vererek, güvenilirlik performansını makul düzeyde tutarken DRL işlem yürütme sayısını azaltmayı sağlamaktadır. Sonuçlar geleneksel optimizasyon tabanlı çözümlere kıyasla eğitim ve çalışma zamanı karmaşıklığında dikkate değer bir azalmayı gösteriyor.

ACKNOWLEDGMENTS

This thesis is not just a reflection of my personal achievement but also a testament to the support and contributions of many wonderful individuals. I want to express my sincere gratitude to the individuals whose constant support and contributions have played a crucial role in the completion of this Ph.D. thesis. Above all, I am deeply thankful to my advisor, Prof. Sinem Çöleri, for her exceptional mentorship, support, and guidance throughout this research journey. I consider myself fortunate to have had the privilege of learning from your visionary mindset and vast experience, which have significantly shaped both my personal and professional growth.

I am also deeply grateful to Prof. Öznur Özkasap, Prof. Ertuğrul Başar, Prof. Özgür Gürbüz, and Prof. Ali Emre Pusane for being part of my dissertation committee. Your valuable feedback and thoughtful comments are instrumental in shaping this dissertation and advancing my research.

I also want to extend my sincere gratitude to Prof. Ahmed Eltawil, Dr. Abdulkadir Çelik, and Dr. Asmaa Abdallah for their invaluable collaboration and mentorship. Their guidance, support, and expertise have significantly contributed to the development of my research, and I am truly thankful for their dedication and insights throughout this journey.

I am deeply indebted to my parents for their unwavering support of my education, for instilling in me strong moral values, and for guiding my personal and academic development. Without your prayers and support, achieving this academic milestone would not have been possible. I am also profoundly grateful to my brothers Yasir Khan and Amir Khan for their constant love, encouragement, and belief in me, which have been instrumental to my success.

Finally, I am truly thankful to my friends and colleagues, for their support and companionship throughout this journey. Your presence made a meaningful difference, and I deeply appreciate each of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

6G	sixth generation
AI	artificial intelligence
BAE	beam alignment engine
BS	base station
CDF	cumulative density function
CNN	convolutional neural network
CSI	channel state information
C-V2X	cellular vehicle-to-everything
DDPG	deep deterministic policy gradient
DFT	discrete Fourier transform
DL	deep learning
DkNN	deep k-nearest neighbors
DQN	deep-Q network
DRL	deep reinforcement learning
DNN	deep neural network
FBL	finite block length
HAP	hybrid access point
IA	initial access
KKT	Karush–Kuhn–Tucker
KPI	key performance indicator
LIME	local interpretable model-agnostic explanation
LSH	locality sensitive hashing
LTE	long-term evolution
MADRL	multi-agent DRL

MINLP	mixed integer nonlinear programming
ML	machine learning
MDP	Markov decision process
MIMO	multiple-input multiple-output
mmWave	millimeter-Wave
NR	new radio
OFDM	orthogonal frequency division multiplexing
O-DFT	oversampled DFT
O-RAN	open radio access network
QoS	quality-of-service
RRM	radio resource management
RSSI	received signal strength indicator
RSU	roadside unit
SNR	signal-to-noise ratio
SHAP	Shapley additive explanations
SFS	sequential feature selection
SLA	service level agreement
TDMA	time-division-multiple-access
UE	user equipment
ULA	uniform linear array
URLLC	ultra-reliable low-latency communication
V2I	vehicle-to-infrastructure
V2N	vehicle-to-network
V2V	vehicle-to-vehicle
VUE	vehicle user equipment
WSN	wireless sensor network
XAI	explainable AI

Chapter 1

INTRODUCTION

Building on the foundation laid by 5^{th} Generation (5G), where Artificial Intelligence (AI) primarily enhances traditional network operations and functionalities, the forthcoming 6^{th} Generation (6G) networks will feature AI-native wireless systems to the full potential by integrating AI/Machine Learning (ML) techniques to intelligently design, optimize, and manage every layer of the communication stack. Recognizing this paradigm shift, the International Telecommunication Union's IMT-2030 framework identifies integrated AI as a foundational pillar of 6G, underscoring the need for AI to be deeply and seamlessly embedded within wireless infrastructures. Despite AI's remarkable performance, lack of transparency and trust in the model decisions is a major bottleneck in its practical deployment for mission-critical services. Developing quantifiable metrics to ensure the confidence and credibility of the results generated by AI-driven approaches is necessary to establish a reliable 6G network in practical scenarios [Hamon et al., 2020]. The widespread adoption of AI techniques necessitates AI/ML algorithms that are robust and explainable to foster human trust and advance AI-native networks.

Assessment of the AI models in a precise context is necessary to meet privacy, fairness, robustness, and explainability criteria. Privacy and fairness are crucial to ensure that AI models are not discriminatory or biased. Sensitive AI applications often require understanding how data is being processed and how the model makes decisions, allowing for transparency and accountability and assuring individuals' privacy is not compromised. However, when it comes to radio resource management (RRM), the focus shifts toward interpreting the inference process of AI models and their robustness against adversarial attacks. It is essential to understand the

model's inner workings to improve its performance, enabling the use of simplified model structures with fewer parameters. This can also help improve the model's robustness to adversarial attacks, increasing the detection accuracy from both the model and data aspects. Robustness is a critical aspect of trustworthy AI, as it provides confidence in the decision outcomes. This confidence can be translated into trust metrics that are understandable to both technical and non-technical audiences. The use of trustworthy AI models in RRM can have significant impacts on performance and security, and the ability to interpret and explain their operations is crucial to their success.

AI-based RRM usually relies on complex deep learning (DL)-based models and algorithms that are hard to comprehend and interpret due to their black-box nature [Guo, 2020]. One of the key drawbacks of DL-based models is the difficulty in understanding how input data features influence the model's output and the logical reasoning processes behind its decisions, causing a lack of interpretation and transparency in the decision-making process. Moreover, AI-based models can be sensitive to biases in the training data, which can lead to poor performance or even discriminatory outcomes when the model is applied to new data that differs from the training data. Additionally, AI-based models can be uncertain about their predictions, making it challenging to understand the confidence level of a given decision. The deficiency of explainability and robustness in AI-based models poses a significant risk of losing control over decision-making beyond the comprehension of wireless system designers and end-users, ultimately resulting in a shortfall of human trust [Rawal et al., 2022].

To incorporate explainability and robustness into AI-based RRM for trustworthy AI, the concept of explainable AI (XAI) has emerged as a crucial tool. XAI encompasses processes and methods that reveal the inner workings of complex AI models, helping to balance the trade-off between explainability and performance inherent to these models. As 6G networks are anticipated to incorporate AI technologies for network automation and optimization heavily, XAI becomes essential to comprehend complex cognitive processes behind AI-driven decision-making, thereby overcoming

challenges of their automated use in network management. Otherwise, the lack of explainability may hinder human trust and slow the adoption of AI-enabled 6G services. Besides, XAI can simplify AI models to reduce the computational and cognitive burden on the systems/engineers. XAI methods, such as Local Interpretable Model-Agnostic Explanations (LIME) [Ribeiro et al., 2016] and SHapley Additive exPlanations (SHAP) [Lundberg et al., 2020], can illuminate the inner workings of complex decision-making processes. LIME is an XAI method that creates a locally faithful representation of the original model behavior by approximating it with a simple, interpretable model. The LIME controls the trade-off between explanation accuracy and its interpretability by introducing a complexity measure. The SHAP method assigns importance values to each feature based on their contribution to the model's predictions, allowing for the exclusion of less significant features without sacrificing performance. SHAP has a strong theoretical foundation based on Shapley values in game theory and exhibits several desirable properties compared to other XAI methods. Radio engineers can manually select or automate the explainable feature selection using these XAI methods, e.g SHAP and LIME, to pinpoint the most influential features affecting the outcomes of the learning task.

XAI techniques can be effectively exploited to improve explainability and robustness of AI-empowered RRM. XAI is gaining significant momentum in wireless communications to aid the understanding of complex AI models [Li et al., 2020], and to conceive new strategies and algorithms for saving communication and computational resources in network design, management, and optimization across all layers of the communication infrastructure. A generic XAI framework for explainable and robust AI-driven RRM, along with relevant techniques and tools to enhance explainability and robustness, is illustrated in Fig.1.1. The relevant XAI techniques supporting efficient AI-empowered RRM are detailed below[1]:

Feature Visualization and Attribution Feature visualization techniques of-

[1]N. Khan, S. Coleri, A. Abdallah, A. Celik and A. M. Eltawil, "Explainable and Robust Artificial Intelligence for Trustworthy Resource Management in 6G Networks," in IEEE Communications Magazine, vol. 62, no. 4, pp. 50-56, April 2024, doi: 10.1109/MCOM.001.2300172.

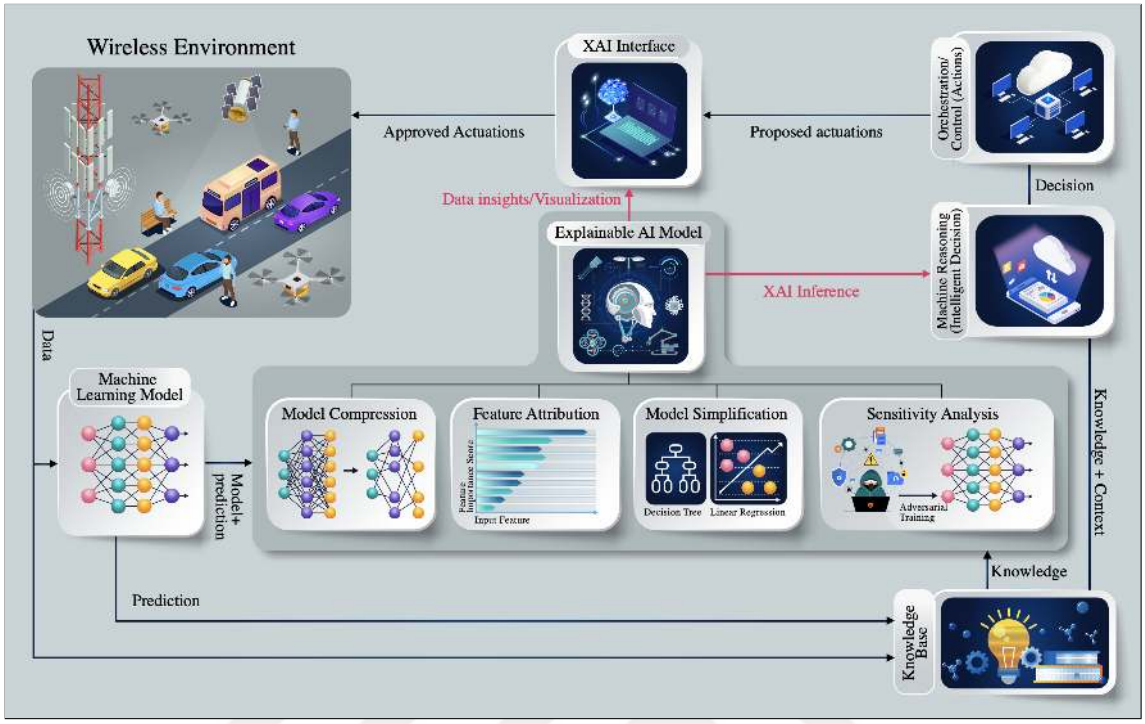


Figure 1.1: XAI-empowered framework for explainable and robust RRM.

fer simple visual illustrations to explain the inner workings and interactions of AI-driven models. Visual augmentation and dimensionality reduction methods can be exploited to generate human-readable interpretations to provide meaningful insights about the model behavior and the decision outcomes. Feature visualization in RRM enables visualizing the characteristics of inputs strongly influencing the output and the derivation of input-output mapping produced by the model, assisting in avoiding conflicts or system-level failures. Feature attribution methods can further enhance interpretability by assigning importance scores to individual features and assessing their impact on the model outcome and prediction performance. SHAP is a particularly effective approach providing a unified framework that assigns each feature an importance value for a particular prediction. Generating feature importance-based explanations for AI-driven RRM problems can reduce the state space for training the DRL agents by filtering the most relevant data to be fed to the AI model, thus simultaneously simplifying the model and improving its understandability.

Model Simplification and Interpretability DNNs have high predictive power but lack interpretability for wireless network operators. XAI methods offer a solution by simplifying DL-based models, maintaining performance, and providing human-understandable interpretations. Surrogate models, e.g., linear regression or decision trees, can simplify the perception of the complex DL-based models and facilitate the interpretability of AI-empowered RRM methods. For instance, the LIME method can be exploited to simplify the DRL-based user traffic offloading in a generic wireless network by dividing the user load demand locally and linearly between the ultra-reliable low-latency communication and enhanced mobile broadband load demands [Guo, 2020]. The usage of surrogate models in RRM makes model decisions easy to interpret while decreasing decision-making latency and resource consumption.

Model Compression Model compression techniques aim to reduce the size and complexity of models while preserving their accuracy, making them more interpretable and easier to understand. Model compression reduces the energy and memory requirements of the model, which can be critical for resource-constrained devices and systems. Compressing DNNs by exploiting sparsity within the network can be achieved by combining model compression techniques (e.g., model pruning, parameter quantization, low-rank decomposition, lightweight model design) with XAI methods such as Deep Learning Important FeaTures (DeepLIFT) [Sabih et al., 2020], a back-propagation-based XAI approach, for obtaining the importance of neurons. Compressing DNNs in DL-based RRM schemes through XAI methods can reduces the communication overhead for the periodic broadcast of model parameters and increases the convergence speed.

Sensitivity Analysis With the ever-increasing adoption of AI-based models for RRM, it is certain that attackers will increasingly seek out new methods of attacking the underlying AI-empowered systems, which has become a critical challenge in many real-world applications. To address this issue, sensitivity analysis can be employed to develop various reactive and proactive defense mechanisms to understand the vulnerabilities of the DNN models and data aspects. Backpropagation approaches, which involve computing the gradient of the prediction output with re-

spect to the features of the input data, can be applied in RRM solutions to unveil the rationale behind AI predictions and improve their robustness. Backpropagation-based techniques such as integrated gradients and layer-wise relevance propagation can be used in AI-based RRM to analyze the impact of changing network parameters, such as the number of antennas, modulation schemes, and transmission power levels on the key performance indicators.

Within the wireless communications literature, research on the XAI application remains limited. Several studies have begun integrating XAI techniques to enhance transparency and understanding within complex network systems [Zhang et al., 2022, Terra et al., 2020, Barnard et al., 2022, Ben Saad et al., 2022]. For instance, classical XAI methods and SHAP have been recommended to analyze the root cause of Service Level Agreement (SLA) violation prediction in a 5G network slicing setup [Terra et al., 2020]. A two-stage pipeline is introduced for robust network intrusion detection using extreme gradient boosting (XGBoost), and SHAP is employed to explain intrusion attacks [Barnard et al., 2022]. Further, XAI is integrated with federated learning to predict and interpret network slices' latency key performance indicator (KPI) [Ben Saad et al., 2022]. The scope of the above works [Terra et al., 2020, Barnard et al., 2022, Ben Saad et al., 2022] does not extend beyond providing insights into the feature relevance in the decision-making process of the underlying machine learning models in different wireless settings. Despite the surge in publications related to XAI techniques in computer vision and surveys emphasizing AI/ML models' explainability and trustworthiness in wireless settings [Brik et al., 2023], the applicability of XAI methods for RRM at the physical and MAC layer remains minimally addressed. The literature lacks a systematic methodology for applying XAI methods to RRM problems, such as model simplification and credibility assessment of decision outcomes for robustness against adversarial/out-of-distribution inputs.

Moreover, model-based deep neural networks (DNNs) aim to benefit from mathematical models that have been developed for years in the wireless communications domain, merging model-based and data-driven approaches to optimize resource management. These DNNs are particularly promising when theoretical models are either

accurate but computationally intractable or tractable but inaccurate [Zappone et al., 2019]. However, their performance hinges on large realistic datasets, which are difficult to obtain in practical systems. As networks grow, the increasing number of DNN input and output ports raises scalability concerns. Additionally, supervised learning methods rely on the precision of the system model used to generate training data, necessitating retraining whenever the system model or key parameters change. Given these limitations, reinforcement learning (RL) has emerged as an effective approach for decision-making under uncertainty, efficiently optimizing complex objective functions. By combining RL with DNNs, deep reinforcement learning (DRL) extends these capabilities to large state-action spaces, offering a powerful solution for real-time resource management [Luong et al., 2019].

Existing DRL-based RRM strategies [Guo et al., 2023, Xiang et al., 2021, Li et al., 2022, Zhang et al., 2020, Ye et al., 2019] may not be suitable for realistic 6G resource allocation problems in future wireless/vehicular systems because the learning process is conducted in a time-triggered mechanism, i.e., the resource allocation decisions are updated periodically in each time frame, requiring computationally intensive training procedures. Against the time-triggered mechanism, event-triggered learning has been proposed with the goal of activating computations or communications only when the system state deviates from the expected accuracy or when the triggering condition is violated [Lu et al., 2023]. Recently, event-triggered learning and DRL have been combined, where the proposed methodology involves incorporating the triggering condition as communication loss into the reward function, leading to more efficient DRL processing by reducing the computation time while providing similar performance [Wang et al., 2020b].

1.1 Thesis Contributions

In this thesis, we extensively study the use of XAI methods for model simplification, robustness assessment, and efficiency improvement in two practical case studies of RRM in 6G networks:

The former case focuses on exploiting XAI to design a novel, robust, and explain-

able DL-based beam alignment engine (BAE) to predict millimeter-wave (mmWave) beams during the initial access (IA) process in MIMO systems. We identify three critical challenges with the existing beam prediction solutions and propose effective methods to resolve these challenges: 1) The performance of the DL-aided approach is highly dependent on received power measurements, which are affected by the quality of the estimated channel measurements. Obtaining large-scale channel datasets from real environments is quite challenging in wireless scenarios due to various channel, antenna, and network topology effects; 2) The accuracy of DL-based beam prediction depends not only on the number of beams used as inputs but also on the specific selection of beams. Identifying and prioritizing important spatial directions is critical to reducing beam sweeping time and improving prediction efficiency. 3) DL-aided approaches struggle to quantify prediction confidence due to their black-box nature, making it difficult to trust their decisions in real-world scenarios, where inaccurate beam selection can significantly degrade performance. To address the first challenge, we propose using a site-specific digital twin (DT) of the wireless environment generated via accurate ray tracing to create synthetic channel data resembling real-world environments. To address the second challenge, we utilize the feature relevance-oriented XAI to identify the most promising beams, further reducing the number of inputs and avoiding unnecessary beam scanning. This reduces overhead while maintaining robust prediction performance. Finally, to address the third challenge, we incorporate the Deep k-nearest Neighbors (DkNN) approach into our beam alignment framework, analyzing the internal representations of the neural network to strengthen the explainability and robustness of its beam predictions.

The latter case focuses on exploiting XAI to simplify the model complexity by reducing the input size of DRL agents for scalable RRM in vehicular networks. Particularly, we propose a distributed multi-agent deep reinforcement learning (MADRL) algorithm to jointly optimize transmit power and spectrum sub-band assignment, enhancing the performance of vehicle-to-network (V2N) and vehicle-to-vehicle (V2V) links in ultra-reliable and low-latency communications (URLLC)-enabled vehicular communications network. Utilizing the SHAP method from XAI, we design a

post-hoc and model-agnostic explainability pipeline that improves understanding of DRL agent inferences and simplifies the agent's inputs through feature selection based on importance rankings. We propose a novel two-stage systematic explainability framework leveraging feature relevance-oriented XAI to simplify the DRL agents. The former stage involves generating state feature importance ranking of the trained models using the SHAP-based importance scores. The latter stage exploits these importance-based rankings to simplify the state space of the agents, whereby the least important features are removed from the model's input, i.e., by masking feature values and observing the effect on the model's performance using an automated novel variable feature selection method.

Additionally, inspired by event-triggered control systems [Lu et al., 2023, Yang et al., 2019], we propose a novel DRL-based framework for the joint power control and block length allocation to minimize the worst-case decoding-error probability in the finite block length (FBL) regime for ultra-reliable low-latency communications (URLLC)-based downlink vehicle-to-everything (V2X) communication system. Initially, an algorithm grounded in optimization theory is developed based on deriving the joint convexity of the decoding error probability in the block length and transmit power variables within the region of interest. Subsequently, an efficient event-triggered DRL-based algorithm is proposed to solve the joint optimization problem. Incorporating event-triggered learning into the DRL framework enables assessing whether to initiate the DRL process, thereby reducing the number of DRL process executions while maintaining reasonable reliability performance.

1.2 Organization

The outline for the rest of the thesis is as follows:

- Chapter 2 describes our designed explainable DL-based beam alignment approach that uses received signal strength indicator (RSSI) measurements from wide sensing beams to predict optimal narrow beams for mmWave MIMO systems efficiently. A model refinement via transfer learning is proposed to

fine-tune the pre-trained model residing in the DT with minimal real-world data, effectively bridging mismatches between the digital replica and real-world environments. Moreover, a model-agnostic explainability framework is proposed to further minimize beam training overhead by leveraging feature relevance through SHAP. It is demonstrated that the proposed approach can significantly reduce beam training overhead compared to the conventional exhaustive search method while providing close-to-optimal performance in terms of accuracy and spectral efficiency.[2]

- Chapter 3 focuses on the robustness aspect and describes a robust DL-based beam alignment framework to predict and filter outlier/adversarial input for efficient and reliable beam management in mmWave MIMO systems. To ensure transparency and robustness, the DkNN approach is utilized, which inspects the internal representations of the model to evaluate how well their predictions conform with the training data, providing interpretable insights into beam selection and robustness against outlier inputs. It is shown that the resulting hybrid beam classifier is model-agnostic and can effectively identify out-of-distribution or outlier inputs without compromising the model's prediction performance.[3]

- Chapter 4 proposes a novel distributed MADRL framework to jointly optimize transmit power and spectrum sub-band assignment, enhancing the performance of V2X communication networks. A novel two-stage systematic explainability framework leveraging the SHAP method from XAI is proposed to simplify the input size for the DRL agents. It is demonstrated that the

[2]N. Khan, A. Abdallah, A. Celik, A. M. Eltawil, and S. Coleri, "Digital Twin-Assisted Explainable AI for Robust Beam Prediction in mmWave MIMO Systems,", in IEEE Transactions on Wireless Communications, July 2025, doi.org/10.48550/arXiv.2507.14180.

[3]N. Khan, A. Abdallah, A. Celik, A. M. Eltawil, and S. Coleri, "Explainable and Robust Millimeter Wave Beam Alignment for AI-Native 6G Networks," in Proceedings of the 2025 IEEE International Conference on Communications (ICC) – Mobile and Wireless Networks Symposium, Montreal, Canada, June 2025.

proposed methodology can simplify the pre-trained models in terms of the average training time, the number of broadcast parameters, and the number of optimal state features required for training without significant performance loss.[4]

- Chapter 5 combines event-triggered learning with reinforcement learning and proposes a novel learning-based framework for the joint power and block-length allocation in a V2X network to ensure URLLC. Initially, an algorithm grounded in optimization theory is developed based on deriving joint convexity of the decoding error probability in the block length and transmit power variables within the region of interest. Subsequently, an efficient event-triggered DRL-based algorithm is proposed to solve the joint optimization problem. Incorporating event-triggered learning into the proposed DRL framework relieves the centralized trainer of excessive computation stress by minimizing the DRL training executions while outperforming the proposed joint optimization method and the recent state-of-the-art DRL-based methods.[5]

- Chapter 6 summarizes our findings, provides potential future research directions, and concludes the study.

[4]N. Khan, A. Abdallah, A. Celik, A. M. Eltawil and S. Coleri, "Explainable AI-aided Feature Selection and Model Reduction for DRL-based V2X Resource Allocation," in IEEE Transactions on Communications, doi: 10.1109/TCOMM.2025.3554655.

[5]N. Khan and S. Coleri, "Event-Triggered Reinforcement Learning Based Joint Resource Allocation for Ultra-Reliable Low-Latency V2X Communications," in IEEE Transactions on Vehicular Technology, vol. 73, no. 11, pp. 16991-17006, Nov. 2024, doi: 10.1109/TVT.2024.3424398.

Chapter 2

XAI-BASED BEAM SELECTION FOR WIRELESS DIGITAL TWINS

2.1 Background and Overview

In the current 5G standard, beam alignment strategies are employed wherein the BS sweeps beams using reference signals, while the UE measures the received signal strength indicators (RSSIs) and reports the strongest beam back to the BS [Xue et al., 2024a, Giordani et al., 2018]. Standard approaches often use quantized beams, which distribute energy across the angular space via codebooks, such as discrete Fourier transform (DFT) codebooks. While DFT codebooks ensure broad coverage, they often lack the granularity for precise beam alignment. To address this, oversampled DFT (O-DFT) codebooks provide finer granularity at the cost of increased beam training overhead, as more beams need to be evaluated during the alignment process. In general, beam sweeping at the base station (BS) or UE side is cyclically executed via an exhaustive search to refresh and maintain continuous beam alignment, where the feedback communication overhead emerges as a critical bottleneck. Employing DL strategies for beam sweeping and beam prediction is a promising means of potentially saving the signaling overhead while providing large latency gains [Xue et al., 2024b].

Generally, the beam sweeping time dominates the overall IA time, as it requires a considerable number of wide sensing beam measurements and reporting to cover the angular space. It is necessary to accurately identify the most effective beam pairs between a BS and the UEs. The accuracy of the DL-based beam prediction task is not solely determined by the number of sensing beams (RSSIs) included in the input but also by the specific combination of beams selected [Cousik, 2022]. In

other words, when the number of DNN inputs is fixed, some combinations of inputs are more effective in accurately predicting the best-oriented beams. It is important to identify the best combination of input beams to increase prediction accuracy while minimizing the total number of beams used. At this point, input feature selection methods can help by offering techniques for feature importance analysis. This aligns with feature selection principles [Chandrashekar and Sahin, 2014], which aim to identify a minimal set of input variables that effectively reduce the objective function. Traditional filter-based feature selection methods rank input variables using criteria such as mutual information or Pearson's correlation, assessing importance based on data characteristics but overlook algorithm dependence and feature interactions. Wrapper methods (e.g., Recursive Feature Elimination) evaluate combinations of input features using the target learning algorithm, scoring each set to select the one with the highest performance, but are computationally expensive for large datasets. XAI-based methods for feature selection have garnered significant attention in wireless communications to determine the primary attributes influencing the model decision and enhancing model performance [Guo, 2020]. XAI-based feature attribution methods, such as SHAP [Lundberg et al., 2020], can highlight the spatial directions or features that most impact the model's predictions. SHAP, a feature attribution method, offers a systematic feature selection strategy by identifying the most impactful features (in our case, RSSI values) influencing the model's predictions. The model is then retrained using this subset of critical features, enhancing understanding while reducing execution time and model complexity (e.g., via feature selection), thus improving the efficiency of the beam alignment process [Khan et al., 2024].

One critical challenge with DL-based data-driven solutions for beam alignment/tracking is the scarcity of real-world data. Training a DL-based beam alignment framework often requires fairly large realistic datasets on the order of 100K-1,000K training samples [Cousik, 2022], [Heng et al., 2022] that are expensive to collect in deployed systems. Recent research exploring the synergies between DTs and wireless communication systems can potentially reduce or even eliminate the real-

world channel acquisition overhead by approximating real-world communications using digital replicas based on accurate 3D models and efficient ray tracing. The digital replica relies on accurate ray tracing to simulate wireless channels using detailed environmental information, including objects' position, dynamics, shapes, and materials, to create high-fidelity 3D models of their surroundings [Alkhateeb et al., 2023]. DT-enabled network systems can achieve precise real-time digital representations of the environment to simulate communication channels and solve real-world communication tasks, including the physical layer reliability aspect of device-to-device links [Zelenbaba et al., 2022], beam prediction tasks [Jiang and Alkhateeb, 2023], channel compression and feedback [Luo and Alkhateeb, 2024], and localization tasks [Morais and Alkhateeb, 2024]. For the beam prediction task, one effective solution is to simulate channels within a digital replica of the wireless environment and fine-tune the BAE with limited real-world data. Using these 3D models, site-specific DTs can generate synthetic channel data that closely mirror real-world conditions. Copies of the trained classifier can be constantly fine-tuned or re-trained using data continuously generated by the DT in the background, which can be rapidly deployed to all the BSs to replace the outdated models and calibrated through few-shot transfer learning [Alkhateeb et al., 2023], [Jiang and Alkhateeb, 2023].

The systematic methodology described in this chapter overcomes the challenges with real-world data collection and efficient beam prediction by integrating digital twins and XAI to select promising sensing beams with minimal beam sweeping overhead for predicting mmWave beams during the IA process. The original contributions of this chapter are listed as follows:

- A DL-based beam alignment framework is proposed that leverages RSSI feedback from a finite set of sensing beams (DFT codebook) to predict optimal narrow beams from the O-DFT codebook for IA and data transmission. By employing a site-specific DT with accurate ray tracing, synthetic datasets resembling real-world environments are generated for model training. Further, an innovative transfer learning-based approach is devised for periodic refine-

ment of the trained model using small amounts of real-world data augmentation, bridging the gap between synthetic and actual data distributions.

- A novel two-stage explainability framework leveraging feature relevance-oriented XAI is proposed to identify the most influential sensing beams. The former stage involves generating feature importance ranking of the model's inputs using the SHAP-based importance scores. The latter stage exploits these rankings to reduce the feature input set to the DNN by removing the least important features from the model's input. The proposed approach reduces beam sweeping overhead by at least 62% while maintaining top-k accuracy comparable to models trained on the full feature set. Moreover, compared to an exhaustive search over all O-DFT beams, the proposed DL-based approach reduces beam training overhead by $\approx 89\%$ while maintaining close to optimal spectral efficiency performance.

2.2 System Model

We consider a narrowband downlink mmWave communication system, where the BS features a uniform linear array (ULA) with N_{BS} antenna elements to communicate with N_{U} single-antenna UEs, as illustrated in Fig. 2.1. We concentrate on the scenario of multi-user beamforming, where the BS communicates with every UE using only a single stream. Notably, for mmWave frequencies, channels often exhibit sparsity in the angular domain, leading to a constrained number of channel paths, L. The narrowband mmWave channel from the BS to the UE_u can be described by a multipath geometric channel model [Heath et al., 2016] as

$$\mathbf{h}_u = \alpha_{u,l}\mathbf{b}\left(\phi_{u,l}\right), \tag{2.1}$$

where $\alpha_{u,l}$ represents the complex path gain for the l-th path of the u^{th} UE, $\phi_{u,l}$ is the angle of departure for the l-th path, and $\mathbf{b}\left(\phi_{u,l}\right)$ represents the beam steering

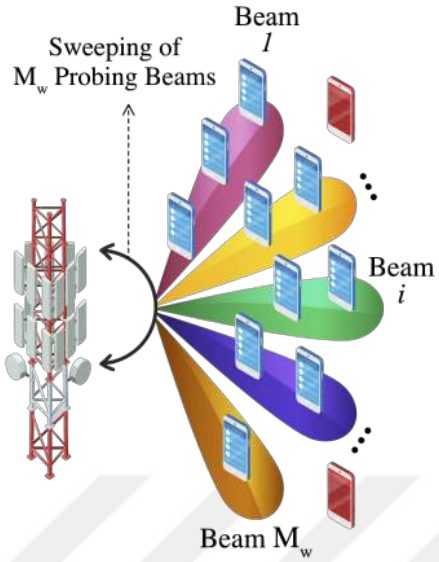


Figure 2.1: System model illustration for the beam alignment in mmWave communication system.

vector, which is given by

$$\mathbf{b}(\phi_{u,l}) = \frac{1}{\sqrt{N_{\mathrm{BS}}}}\left[1, e^{j\frac{2\pi}{\lambda}d\sin(\phi_{u,l})}, \ldots, e^{j(N_{\mathrm{BS}}-1)\frac{2\pi}{\lambda}d\sin(\phi_{u,l})}\right]^T, \tag{2.2}$$

where λ is the signal wavelength given by $\lambda = \frac{c_0}{f_0}$, c_0 is the speed of light, f_0 is the carrier frequency, and $d = \frac{\lambda}{2}$ denotes the antenna spacing. Note that for the narrowband channel model, we made the implicit assumption (very common in most beamforming and array processing literature) that the spatial directions are frequency-independent and the communication bandwidth B is much smaller than the carrier frequency f_0, such that the array response is essentially constant with $f \in [f_0 - B/2, f_0 + B/2]$. To mitigate the hardware cost and power consumption of a fully digital system, we adopt analog-only beamforming where the BS has a single common transmit/receive radio frequency (RF) chain shared by N_{BS} antennas. Hence, the beamforming vector at the BS is given by

$$\mathbf{w} = \frac{1}{\sqrt{N_{\mathrm{BS}}}}\left[e^{j\varphi_1}, \ldots, e^{j\varphi_i}, \ldots, e^{j\varphi_{N_{\mathrm{BS}}}}\right]^\top \in \mathbb{C}^{N_{\mathrm{BS}}\times 1}, \tag{2.3}$$

where φ_i is the phase shift of i-th antenna element. We assume that the BS adopt a beamforming codebook $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_Q\}$ incorporating Q pre-defined beamform-

ing vectors. The vectors in $\mathbf{W}$ satisfy $\|\mathbf{w}_q\|^2 = 1, \forall q \in \{1, \ldots, Q\}$ to accommodate the constant-modulus constraint of analog beamforming architecture.

During the beam sweeping process, the BS periodically transmits symbols $s_w \in \mathbb{C}$ to the UEs through the beams defined by the matrix $\mathbf{W} \in \mathbb{C}^{N_{\text{BS}} \times Q}$. Following the beam sweeping process, the complex received signal at the UE_u, using the q_u-th beamforming vector $\mathbf{w}_{q_u}$, can be expressed as

$$r_u = \sqrt{P_{\text{BS}}} \mathbf{h}_u^H \mathbf{w}_{q_u} s_w + z_u, \tag{2.4}$$

where P_{BS} denotes the BS transmit power, $\mathbf{h}_u \in \mathbb{C}^{N_{\text{BS}} \times 1}$ is the channel vector, and z_u represents the additive complex noise with power σ_z^2. Then, with unit-power transmitted symbols, the signal-to-noise ratio (SNR) at the UE_u can be expressed as

$$\text{SNR}_u = \frac{P_{\text{BS}} \left| \mathbf{h}_u^H \mathbf{w}_{q_u} \right|^2}{\sigma_z^2}. \tag{2.5}$$

2.3 Problem Formulation and Solution Methodology

This section describes the problem formulation for the beam alignment task and introduces our proposed DT-aided explainable and robust beam alignment framework.

2.3.1 Problem Formulation

For a given BS-UE pair, the optimal beam index q_u^* can be identified by selecting the beam that maximizes the SNR:

$$\begin{aligned} q_u^* &= \underset{q_u \in \{1,2,\ldots,Q\}}{\arg\max} \left(\frac{P_{\text{BS}} \left| \mathbf{h}_u^H \mathbf{w}_{q_u} \right|^2}{\sigma_z^2} \right) \\ &= \underset{q_u \in \{1,2,\ldots,Q\}}{\arg\max} \left(\left| \mathbf{h}_u^H \mathbf{w}_{q_u} \right|^2 \right). \end{aligned} \tag{2.6}$$

Notably, the overall number of beams swept remains constant, unaffected by the number of UEs, as multiple UEs can measure the sensing beams concurrently, utilizing shared time and frequency resources. It is worth noting that identifying

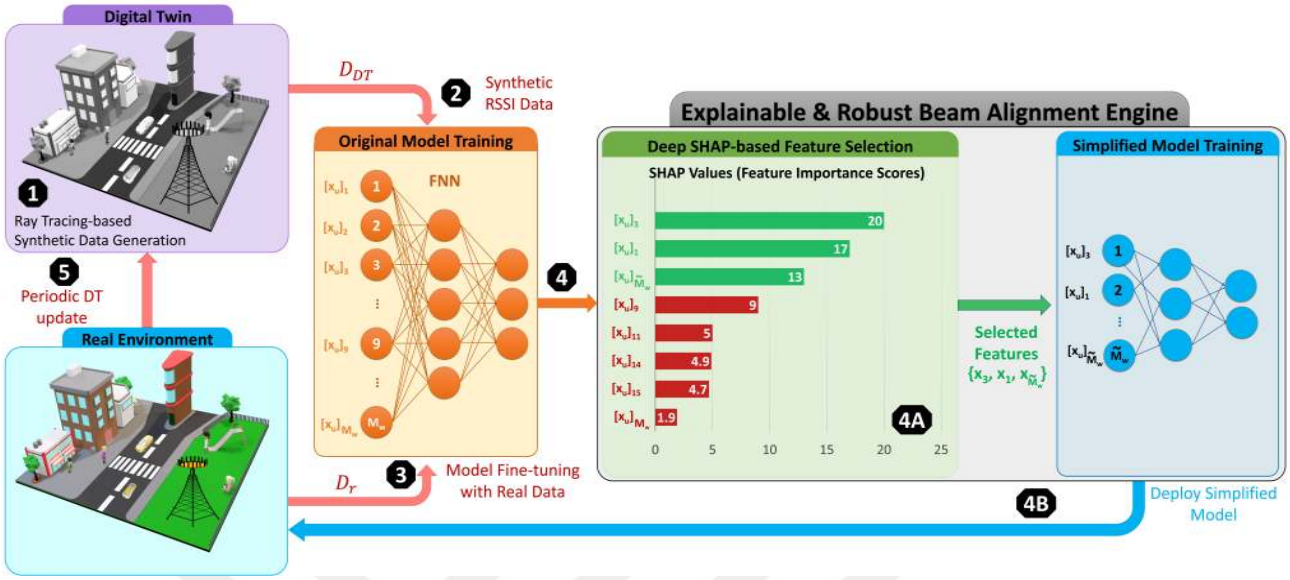


Figure 2.2: Illustration of the proposed DT-aided explainable and robust beam alignment framework.

q_u^* through exhaustive beam search over Q codewords results in significant beam sweeping overhead. To address this issue, we leverage the exploration capabilities of DL to identify optimal beam directions tailored to specific sites.

2.3.2 Solution Methodology

To overcome the challenges with existing DL-based beam alignment solutions in terms of lacking real training datasets, high beam training overhead, and lack of transparency in decision-making, we propose a framework where the BS uses the digital replica of the physical environment to enhance network performance and provide an explainable and robust solution for the beam alignment task. As illustrated in Fig. 2.2 and introduced in the sequel sections, the system operation proceeds as follows:

① Synthetic channel data is first generated in a site-specific DT using ray tracing, [c.f. Section 2.4.2].

② The synthetic datasets are then utilized to train the DL-based BAE model,

[c.f. Section 2.4.2].

③ The DL-based BAE, initially trained on synthetic data, is then fine-tuned by augmenting it with a small portion of real-world data. This enhances the generalization capability of the DT models and mitigates the distribution gap between real and synthetic data, [c.f. Section 2.4.3].

④ The post-hoc explainability and robustness assessment then begins with Ⓐ, where the deep SHAP method from XAI is employed to assign importance scores to the input features. This enables the selection of a subset of features, such as RSSIs, that significantly contribute to the DL model's predictions [c.f. Section 2.5]. Then in Ⓑ, the simplified model is trained with the selected input features and deployed in a real-world environment for online operation.

⑤ The DT is periodically updated where the BS collects real data through user feedback during deployment to compensate for impairments in the DT data and synchronize the DL model, e.g., when the environment experiences significant changes, by repeating ①, ②, ③, and ④.

2.4 Deep learning framework for beam alignment

This section describes the details of the DL-based beam alignment strategy and the DT-based data acquisition. We first describe the beam selection problem as a nonlinear mapping from the collected RSSIs values to the optimal narrow beam index. Then, we detail the DT-assisted data acquisition approach, followed by the transfer learning-based model refinement strategy to overcome the modeling mismatches between the physical environment and its digital replica.

2.4.1 DL-based Beam Alignment

The proposed DL-aided BAE system falls into the beam sweeping framework compatible with the 5G NR standard. The proposed solution leverages the RSSI values

over a small set of compact wide sensing beams M_{w} (i.e., beams from DFT codebook with $M_{\mathrm{w}} \ll Q$), to select the best narrow beam from the O-DFT codebook of size Q, thus, avoiding exhaustive extensive search over the O-DFT codebook. For beam-sweeping, the BS transmits the pilot signals over a smaller set of beams, each at a separate time slot. All UEs connected to the BS measure and report the RSSI values of the M_{w} beams. It is assumed that beam sweeping, measurement, and reporting occur within the coherence time during which the channel remains constant. The reported beam sweeping results for UE u can be written as

$$\mathbf{x}_u = [[x_u]_1, \ldots, [x_u]_{M_{\mathrm{w}}}] = \left[|[r_u]_1|^2 \cdots \left|[r_u]_{M_{\mathrm{w}}}\right|^2 \right]^T, \tag{2.7}$$

where $[r_u]_i = \sqrt{P_{\mathrm{BS}}}\mathbf{h}_u^H \mathbf{w}_i s_w + z_i$ is the received signal using the i-th sensing beam, $\forall i \in M_{\mathrm{w}}$.

In the proposed DL-based approach, a DNN is trained to associate the correct/best-oriented beam between the BS and UE with the RSSIs over wide sensing beams as the DNN's inputs and the index of the corresponding optimal narrow beam as the output. This beam selection problem is formulated as a classification task, where each classified category corresponds to one candidate narrow beam. Particularly, the reported beam sweeping results from (2.7) are fed as input to DNN-based beam classifier denoted by $f(\cdot;\boldsymbol{\theta}) : \mathcal{X} \rightarrow \mathbb{R}^Q$, where $\mathcal{X} \in \mathbb{R}^{M_{\mathrm{w}}}$ is the set of DNN's input corresponding to the RSSI values over M_{w} beams and $\boldsymbol{\theta}$ denotes the DNN's weight parameters vector.

One challenge with the DL approach is the demand for real-world training data. The DL model typically requires a dataset that captures the environmental characteristics. For the beam alignment task in (2.6), the goal is to assign the optimal beam index by identifying the spatial directions that provide the strongest RSSI values, which are a function of channel characteristics $\mathbf{h}_u$, which in turn depend on the real environment ($\mathcal{E}_u$) for the UE u, i.e., $\mathbf{h}_u = g(\mathcal{E}_u)$, where $g(\cdot)$ represents the wireless propagation model [Alkhateeb et al., 2023]. However, gathering real-world channel measurements is often a challenging and resource-intensive process that constitutes the main obstacle to deploying DL-based BAE in practice.

As a solution to this problem, transfer learning techniques can be used to transfer parts of information in a network previously trained with a large dataset to another network for which only a small training dataset is available [Zhuang et al., 2021]. This process, known as domain adaptation, enables information to be shared across different environments. Assuming a large dataset of synthetic measurements can be collected from a digital environment, which we refer to as source domain (DT scenario), the DNN weights can be trained by a standard backpropagation algorithm. On the other hand, in a second environment, which we refer to as the destination domain (real scenario), only a limited-size dataset is available since the destination domain corresponds to real-world deployment. Then, the information learned from the source domain can be transferred to the destination via transfer learning [Rezaie et al., 2021], allowing the model to learn from previously encountered data and new data samples. Next, we detail the procedure for channel data acquisition and synthetic dataset construction by leveraging the site-specific DT to approximate the communication environment.

2.4.2 Site-specific DT-aided Data Acquisition

As illustrated in Fig. 2.2, site-specific DT resides at the BS to generate synthetic channel data via ray tracing for learning and optimizing beam selection. This twin can model the communication environment $\mathcal{E}_u$ and the wireless propagation model $g(\cdot)$ through an electromagnetic (EM) 3D model $\mathcal{E}_{u,\mathrm{T}}$ and a ray tracing based approximation $g_{\mathrm{T}}(\cdot)$ of the wireless propagation model [Alkhateeb et al., 2023]. The EM 3D model provides estimated data on the positions, shapes, and materials of the BS, UE, and surrounding objects, whereas ray tracing generates propagation paths using this data to create spatially consistent channels in the digital replica. The generated synthetic channel data from the DT mirrors the distribution of real-world data. In particular, the environment DT constructs a synthetic labeled dataset $\mathcal{D}_{\mathrm{T}} = \left\{(\mathbf{X}_{\mathrm{T}}, \mathbf{q}_{\mathrm{T}}) = (\mathbf{x}_{u,d}, q^{\star}_{u,d}) : d = 1, \cdots, D_{\mathrm{T}}\right\}$, comprising D_{T} samples, where $\mathbf{x}_{u,d}$ represents the RSSI values as input features for the d-th sample, $q^{\star}_{u,d}$ is the O-DFT beam index with the highest RSSI as the target label, generated using

(2.8). Substituting the 3D model $\mathcal{E}_{u,\mathrm{T}}$ and the ray tracing approximation of channel $\mathbf{h}_{u,\mathrm{T}} = g_{\mathrm{T}}(\mathcal{E}_{u,\mathrm{T}})$ into (2.6), the beam prediction task performed in the digital replica can be expressed as

$$q_u^\star = \underset{q_u \in \{1,\ldots,Q\}}{\arg\max} \left| \mathbf{w}_{q_u}^H \mathbf{h}_{u,\mathrm{T}} \right|^2 . \tag{2.8}$$

Furthermore, cross-entropy loss, which quantifies the dissimilarity between predicted probabilities and true labels, is applied to train the DNN's weight parameters, using the one-hot encoding of the optimal beam index in (2.6) as the ground truth label. To facilitate the loss function design during training, we obtain the ground-truth label by searching the optimal beam direction for each sample in the 0-DFT codebook by performing noise-free exhaustive search. The loss function is formulated as:

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_T) = -\frac{1}{D_{\mathrm{T}}} \sum_{d=1}^{D_{\mathrm{T}}} \sum_{q_u=1}^{Q} p_{u,d,q_u}^\star \log \widehat{p}_{u,d,q_u}, \tag{2.9}$$

where $p_{u,d,q_u}^\star$ is the true distribution over labels indicating that the d-th data sample belongs to q_u-th class, such that $p_{u,d,q_u}^\star = 1$ if the q_u-th candidate narrow beam is the actual optimal beam ($q_u^\star$) and $p_{u,d,q_u}^\star = 0$ otherwise, and $\widehat{p}_{u,d,q_u} \in \mathbb{R} \to (0,1)$ represents the predicted probability distribution, obtained by applying a softmax function to the logits produced by the beam classifier.

In practice, constructing a DT that perfectly mirrors the real-world environment is inherently challenging as the DT model approximating the actual environment is prone to distribution shifts and inaccuracies in parameter estimations, often leading to degraded beam alignment accuracy. These inaccuracies arise from the inherent modeling impairments of the DT, which limit the performance of models trained exclusively on synthetic data. As a remedy, a small amount of real-world data augmentation is utilized to fine-tune the pre-trained model (i.e., transfer learning) to transcend the digital replica.

2.4.3 Transfer Learning for Model Refinement

As illustrated in Fig. 2.2, the DT keeps a dynamically updated representation of the actual environment. Copies of the pre-trained DNN are further refined by re-training the pre-trained model using a small amount of real-world data augmentation to achieve similar performance as the model solely trained on real-world data. In particular, transfer learning is leveraged to adapt the model previously trained on the large synthetic dataset $\mathcal{D}_{\mathrm{T}}$ and retrained by augmenting training samples from a real-world dataset. The real-world data, for instance, can be collected from the feedback provided by the UEs during the online operations.

While the training samples in $\mathcal{D}_{\mathrm{T}}$ can be continuously generated by the DT in the background, the BS keeps collecting the RSSI values as feedback from the users to construct the real dataset $\mathcal{D}_{\mathrm{R}} = \{(\mathbf{X}_{\mathrm{R}}, \mathbf{q}_{\mathrm{R}})\} = (\mathbf{x}_{u,d}, q^{\star}_{u,d}) : d = 1, \cdots, D_{\mathrm{R}}\}$, defined similarly as $\mathcal{D}_{\mathrm{T}}$ with size D_{R}. Then, by augmenting a small proportion, e.g., $20 - 30\%$ of real data samples from $\mathcal{D}_{\mathrm{R}}$, the DL-based BAE is refined to achieve comparable performance to the model purely trained on real-world data $\mathcal{D}_{\mathrm{R}}$. Therefore, the model refinement process aims to minimize the loss function in (2.9) on the augmented dataset $\mathcal{D} = \mathcal{D}_{\mathrm{R}} \cup \mathcal{D}_{\mathrm{T}} = \{(\mathbf{X}, \mathbf{q})\} = (\mathbf{x}_{u,d}, q^{\star}_{u,d}) : d = 1, \cdots, D\}$ comprising D data samples instead of D_{T} samples.

2.5 XAI Guided Beam Training Reduction

The DL-based beam alignment strategy described in the previous section relies on sweeping sensing beams at the BS to collect RSSI values (2.7), which are used as input features to the DL model. To reduce sweeping overhead and improve alignment efficiency, we propose an XAI-based feature selection approach to identify a subset of input RSSIs. This approach leverages the fact that certain RSSIs contribute more significantly to the beam selection process, enabling a reduction in beam training overhead by avoiding sweeping the entire set of sensing beams. This section introduces an XAI-based method for optimizing sensing beam selection by analyzing the contributions of RSSI values across candidate beams. Using SHAP [Lundberg and

Lee, 2017], a robust framework for computing local and global feature importance in machine learning models, we assign importance scores to the inputs of a pre-trained model. These scores are then utilized to derive an efficient feature selection strategy, minimizing the *beam sweep time* required during IA. It is worth mentioning that the feature selection process is conducted offline and in a post-hoc manner, meaning that the pre-trained model trained on the augmented dataset $\mathcal{D}$ is readily available for SHAP-based analysis.

As illustrated in Fig. 2.2, SHAP is used for the selection of a subset of RSSI inputs collected over M_{w} sensing beams facilitating an understanding of feature significance. Instead of measuring the RSSI values $\mathbf{x} = \{x_1, \ldots, x_{\mathrm{M_w}}\}$, where each x_i corresponds to the RSSI measurements from the i-th sensing beam, a subset of these RSSI measurement is used to represent the subset of features, i.e, the RSSI values, deemed most important for the model's prediction. We denote the size of SHAP-based selected input features as $\widetilde{M}_{\mathrm{w}}$ such that $\widetilde{M}_{\mathrm{w}} \leq M_{\mathrm{w}}$. Specifically, we aim to identify the most influential features for beam prediction to maximize accuracy while minimizing the number of sensing beams used. For $\widetilde{M}_{\mathrm{w}} = 32$, cycling through all possible combinations would require checking for $2^{32} = 4.29 \times 10^9$ subset of input features to identify the one that performs the best, rendering exhaustive evaluation impractical. For simplicity, we drop the index u in subsequent notations. In the following, we briefly introduce Shapley values, describe SHAP, and detail the proposed XAI-based feature selection algorithm for sensing beam optimization.

2.5.1 Preliminaries on Shapley Values and SHAP

Shapley values quantify the marginal contribution of each input feature x_i, $\forall i \in \{1, \ldots, M_{\mathrm{w}}\}$, to a model's prediction $f(\mathbf{x})$, where x_i is the i^{th} element of input $\mathbf{x}$ (2.7). The Shapley value for feature x_i is computed by evaluating all possible coalitions of features and averaging the weighted differences in predictions with and without x_i. For a prediction model $f(\cdot)$, the Shapley value $\psi_i(\mathbf{x})$ is defined as:

$$\psi_i(f, \mathbf{x}) = \sum_{\mathbf{z}' \subseteq \mathbf{x}'} \frac{|\mathbf{z}'|!(M_{\mathrm{w}} - |\mathbf{z}'| - 1)!}{M_{\mathrm{w}}!} \left[f(\mathbf{z}') - f(\mathbf{z}' \setminus \{x_i\}) \right], \tag{2.10}$$

where $\psi_i(f, \mathbf{x})$ represents the Shapley value for feature x_i, quantifying its contribution to the model's output, M_{w} is the total number of features, $\mathbf{z}'$ is a subset of features, $|\mathbf{z}'|$ denotes the number of original features included in the subset $\mathbf{z}'$, $\mathbf{x}'$ is the set of all original features, $f(\mathbf{z}')$ is the prediction on $\mathbf{z}'$, and $f(\mathbf{z}' \setminus \{x_i\})$ is the prediction without x_i. The term $\frac{|\mathbf{z}'|!(M_{\mathrm{w}}-|\mathbf{z}'|-1)!}{M_{\mathrm{w}}!}$ is the Shapley weight, which ensures a fair attribution by averaging over all possible feature permutations. The Shapley value $\psi_i(f, \mathbf{x})$ measures x_i's contribution to the transition from the model's expected output (without x_i) to the actual prediction. These values include both magnitude and sign, indicating the significance and direction of each feature's impact.

Exact computation of Shapley values requires evaluating all $2^{M_{\mathrm{w}}}$ feature combinations, which is computationally infeasible for DL models with a large number of inputs. To address this, Deep-SHAP offers an efficient approach by combining Shapley values with DeepLIFT, a method that decomposes the output prediction of a neural network by backpropagating contributions of all neurons in the network to the input features [Lundberg and Lee, 2017, Shrikumar et al., 2017]. Deep-SHAP leverages DeepLIFT's per-node attribution property to aggregate Shapley values computed for smaller network components into values for the entire DNN. Deep-SHAP estimates feature importance by calculating the difference between input features and their corresponding reference values, which are derived from a *background* dataset (a subset of the training data). These reference values are propagated through the DNN to precompute reference input and output values for each neuron. When explaining a specific prediction, each input feature is assigned a score that quantifies how its deviation from the reference contributes to the difference in the output. By integrating over multiple samples from the background dataset, Deep-SHAP efficiently approximates Shapley values without the exponential computational cost of evaluating all feature combinations [Shrikumar et al., 2017]. This allows for scalable

and interpretable explanations of DL model predictions.[1]

2.5.2 Deep-SHAP-based Feature Selection Methodology

Deep-SHAP based feature importance analysis requires three inputs: a pre-trained DL model, a background dataset, and a set of data points to be explained. In our analysis, a proportion of the dataset used to train the DL model is used as the *background* dataset $\mathcal{D}_{\text{BG}}$ and the data samples to be explained come from the *test* dataset $\mathcal{D}_{\text{test}}$. As described previously, the dataset $\mathcal{D}_{\text{BG}}$ is used to compute baseline predictions, against which the contributions of individual features in the test dataset are measured [Lundberg and Lee, 2017]. Our proposed methodology consists of two stages: (1) generating feature importance rankings using Deep-SHAP and (2) selecting a reduced subset of input features based on the generated rankings.

In the first stage, the Deep-SHAP explainer from the SHAP library [Lundberg et al., 2020] is instantiated using the pre-trained model $f(\mathbf{x};\boldsymbol{\theta})$ and dataset $\mathcal{D}_{\text{BG}}$, which represents the baseline reference values for feature importance calculations. For each test sample in $\mathcal{D}_{\text{test}}$, the explainer evaluates the contribution of individual features to the model's prediction and outputs the SHAP values signifying the features importance scores.

Formally, Deep-SHAP produces local explanations as importance scores in the form of SHAP values $\boldsymbol{\psi}(\mathbf{x}) = \left[\psi^d_{1,1}, \ldots, \psi^d_{i,q}, \ldots, \psi^d_{M_{\text{w}},Q}\right]$, where $\psi^d_{i,q}$ denotes the impact of feature x_i on the observed output q for data sample d, quantifying the contribution of feature x_i to q-th beam class label. Here, $i \in \{1, \ldots, M_{\text{w}}\}$ indexes the features, $q \in \{1, \ldots, Q\}$ indexes the beam labels, and $d \in \{1, \ldots, D_{\text{test}}\}$ indexes the data samples to be explained. Notice that Deep-SHAP provides SHAP values for each sample instance d and each probable beam index (output class label). Global explanations are obtained by averaging the absolute SHAP values across all data

[1]For more details, we refer interested readers to [Lundberg and Lee, 2017, Shrikumar et al., 2017].

Algorithm 1: Deep-SHAP based Feature Selection

Input: $f(\mathbf{x};\boldsymbol{\theta})$, $\mathcal{D}_{\mathrm{BG}}, \delta$

Output: Subset of input features $\mathbf{x}_S$

Stage 1: Beam importance rankings

  Instantiate the Deep-SHAP explainer;

  Compute $\boldsymbol{\psi}(\mathbf{x}) = \left[\psi^d_{1,1}, \ldots, \psi^d_{\mathrm{M_w},Q}\right]$;

  Aggregate SHAP values as per (2.11);

  $\bar{\boldsymbol{\psi}}(\mathbf{x}) \leftarrow \text{argsort}\left(\left[\bar{\psi}_1, \ldots, \bar{\psi}_{\mathrm{M_w}}\right], \text{descending}\right)$

Stage 2: Threshold-based Feature Selection

  Compute $\mathcal{M}_{\mathrm{SHAP}}(\delta)$ using (2.12) ;

return *Subset of input features* $\mathbf{x}_S$

samples and beam labels:

$$\bar{\psi}_i = \frac{1}{|D_{\mathrm{test}}| \cdot Q} \sum_{d=1}^{|D_{\mathrm{test}}|} \sum_{q=1}^{Q} |\psi^d_{i,q}|, \tag{2.11}$$

resulting in a vector of mean absolute SHAP values $\bar{\boldsymbol{\psi}}(\mathbf{x}) = \left[\bar{\psi}_1, \ldots, \bar{\psi}_i, \ldots, \bar{\psi}_{M_{\mathrm{w}}}\right]$. Sorting this vector in descending order provides the global feature/beam importance rankings, where the first position represents the most important feature. These rankings guide feature selection and can be visualized to provide network operators with interpretable explanations of influential beam patterns.

In the second stage, we devise a feature selection strategy based on the relative significance of SHAP values, i.e., selecting all input features whose importance scores lie above a threshold δ, such that the sum of the SHAP values for the selected features accounts for a fraction δ of the aggregated SHAP values. Mathematically, the selection criterion for feature x_i can be expressed as:

$$\mathcal{M}_{\mathrm{SHAP}}(\delta) = \left\{ i \mid \sum_{i \in \mathbf{x}_S} \bar{\psi}_i \geq \delta \cdot \sum_{j \in \mathbf{x}} \bar{\psi}_j, \right\}, \tag{2.12}$$

where $\mathbf{x}_S$ represents the subset of important features in descending order of SHAP values., i.e, the RSSI inputs corresponding to the highest SHAP values, $\mathbf{x} = \{x_1, \ldots, x_{\mathrm{M_w}}\}$

represents the set of all features, δ is the threshold percentage. Here, $\mathcal{M}_{\text{SHAP}}(\delta)$ is the index set of selected features satisfying the threshold criterion. Using this index set, the subset of features $\mathbf{x}_S = \{x_i \mid i \in \mathcal{M}_{\text{SHAP}}(\delta)\}$ is formed, containing $\widetilde{M}_{\text{w}}$ features, where $\widetilde{M}_{\text{w}} = |\mathbf{x}_S|$. Since the SHAP values are typically concentrated among the top-ranked features (following a diminishing contribution trend), moderately high values of δ (e.g., $> 70\%$), ensure that the most influential features are retained.

The Deep-SHAP based feature selection process is summarized in Algorithm 1. Algorithm 1 provides a systematic method to identify the optimal subset of RSSIs associated with the $\widetilde{M}_{\text{w}}$ sensing beams. The inputs of the algorithm are the pre-trained model to be explained $f(\mathbf{x}; \boldsymbol{\theta})$, the background dataset $\mathcal{D}_{\text{BG}}$, and a feature selection threshold δ. The output is a subset of RSSI measurements $\mathbf{x}_S = \{x_1, \ldots, x_{\widetilde{\text{M}}_{\text{w}}}\}$ selected according to the SHAP importance criterion in (2.12). A simplified model is then retrained in the DT using this subset of critical features $\mathbf{x}_S$, reducing model complexity (e.g., via feature selection) and beam sweeping time by minimizing the number of sensing beams to $\widetilde{M}_{\text{w}}$ instead of M_{w}.

2.6 Simulation Results

In this section, we provide details of the simulation setup, dataset acquisition, and the DL model architecture, followed by a discussion of the results.

2.6.1 Simulation Setup

Scenario Setup

To simulate both the *target scenario* (destination domain) and the *DT scenario* (source domain), we utilize Wireless InSite software [Remcom,] to conduct precise ray-tracing simulations. Wireless InSite models the physical attributes of real-world objects, including their size, location, and material properties, to generate detailed 3D ray-tracing maps of the communication environment. This software is employed for both scenarios to ensure consistent and realistic modeling of mmWave communications.

In the *target scenario*, representing a real-world deployment, we simulate mmWave BS-UE communications within an urban environment modeled after the downtown sector of Boston. The channel dataset is generated using the Boston5G scenario from DeepMIMO [Alkhateeb, 2019], which offers a high-fidelity simulated outdoor environment constructed with Wireless InSite's ray-tracing. This scenario includes a BS, a defined service area, and detailed representations of buildings and foliage, accurately capturing the propagation characteristics of an urban city landscape. The environment comprises both line-of-sight (LOS) and non-line-of-sight (NLOS) users. The BS employs a ULA with $N_{\mathrm{BS}} = 32$ antennas and is positioned at a height of 15 m, oriented towards the negative y-axis. UE devices are modeled as single-antenna systems at a height of 2 m. The service area spans $200\,\mathrm{m} \times 230\,\mathrm{m}$ and is discretized into a user grid with a spacing of 0.37 m. To enhance the stability and efficiency of training, the channel vectors $\mathbf{h} \in \mathbb{C}^{N_{\mathrm{BS}} \times 1}$ are normalized by the largest absolute value in the channel matrix.

The *DT scenario* (source domain) approximates the real-world environment through a *digital replica*, which can be dynamically updated based on live monitoring of the environment and UE distribution. To generate the DT scenario, we adopt the approach described in [Alkhateeb, 2019] and construct the channel matrices using the DeepMIMO dataset generator [DeepMIMO,]. Particularly, the DeepMIMO dataset generator is completely defined by 1) ray-tracing scenario and 2) dataset parameters. In the considered DT scenario setup, we adopt the "Boston5G" ray-tracing scenario [Alkhateeb, 2019]. To account for the approximation errors in the EM 3D models and the difficulty of representing *foliage* due to seasonal variations and unpredictable growth patterns, controlled imperfections are introduced into the DT scenario: foliage is completely omitted, and building positions are randomly shifted by 2 m on the X-Y plane. The dataset parameter adhere to the parameters and system configurations used for the target scenario, summarized in Table 2.1.

Table 2.1: Hyperparameters for channel generation

Name of scenario	**Boston-5G**
Active BS	1
BS transmit power	30 dBm
Active users	1-622
Number of antennas (x, y, z)	(1, 32, 1)
Carrier frequency	28 GHz
System bandwidth	500 MHz
Antenna spacing	0.5
OFDM sub-carriers	1
OFDM sampling factor	1
OFDM limit	1
$D_{\mathrm{T}}, D_{\mathrm{R}}, D_{\mathrm{BG}}$ size	57144, 24485, 8162

Dataset Acquisition

The BS performs analog beamforming with $M_{\mathrm{w}} = 32$ sensing beams using a N_{BS}-DFT codebook, whereas, for the narrow beam codebook $\mathbf{W}$, we use an O-DFT with oversampling factor of 4 to get a total of 128 narrow beams. The parameters of the neural network $\boldsymbol{\theta}$ are learned from the augmented dataset $\mathcal{D} = \mathcal{D}_{\mathrm{R}} \cup \mathcal{D}_{\mathrm{T}}$. In all experiments, 70% of the data is allocated for training, 10% for validation, and the remaining 20% for testing.

All simulations are performed on a 10-Core Intel(R) Xenon(R) Silver 4114, 2.2GHz system equipped with an Nvidia Quadro P2000 graphics processing unit (GPU).

2.6.2 Deep Learning Model Architecture

To show that the proposed feature selection and the robustness assessment methodologies are model agnostic, we consider two different model architectures: 1) A convolutional neural network (CNN) based classifier with $L = 4$ layers with three convolution layers stacked with a fully connected layer. Each convolution layer is

Table 2.2: Architecture and training hyperparameters for DNN and CNN models

Model	**Layer**	**Details**	**Activation**
DNN	Input Layer	Neurons = # RSSI values	-
	Hidden Layer 1	64 Neurons	ReLU
	Hidden Layer 2	64 Neurons	ReLU
	Hidden Layer 3	128 Neurons	ReLU
CNN	Conv Layer 1	Filters = 32, Kernel = 3×3, Stride = 1, Padding = 1	ReLU
	Conv Layer 2	Filters = 64, Kernel = 3×3, Stride = 1, Padding = 1	ReLU
	Conv Layer 3	Filters = 128, Kernel = 1×1, Stride = 1, Padding = 0	ReLU
	Fully Connected	128 Neurons	-
Optimizer		Adam, Learning rate = 10^{-3}	
Loss Function		Cross Entropy	
Epochs		100	

followed by the rectified linear unit (ReLU) activation to provide non-linearity to the convolutional layers. 2) A fully connected DNN classifier with $L = 3$ fully connected layers and using ReLU activation, which takes the RSSI values as input features, i.e., the input neurons equal to the number of RSSI values, followed by three hidden layers. The output layer has neurons corresponding to the possible O-DFT indices and outputs the softmax probability distribution. For a quick nearest neighbor search on DkNN, we use LSH from the FALCONN Python library, which implements cross-polytope LSH and offers sublinear query time performance [Andoni et al., 2015]. We employ a grid parameter search to configure the number of nearest neighbors to $\widetilde{k} = 10$. The hyperparameters for the neural network architectures are summarized in Table 2.2.

2.6.3 Baselines and Metrics

For comparison purposes, we consider the following benchmark methods:

1. *SVD perfect channel* The upper bound beamforming based on the singular value decomposition (SVD) of perfectly known channels under 3-bit quantized phase shifter [Abdallah et al., 2024]. It is worth mentioning that the SVD

upper bound can only be reached when each user's perfect channel is known at the BS.

2. *Exhaustive search O-DFT* (with oversampling factor of ×4): The BS exhaustively sweeps the O-DFT codebook to scan $N_{\mathrm{BS}}\times 4$ narrow beams, and selects the beam with the highest RSSI value for downlink transmission.

3. *2-Tier Hierarchical Beam Search:* The BS employs a two-tier hierarchical codebook, where the tier-1 codebook comprises multiple wide beams, and the tier-2 codebook includes all the narrow beams from the adopted O-DFT codebook of size Q. Each narrow beam in the tier-2 codebook is a child beam associated with a wide beam from the tier-1 codebook. The BS first sweeps the $M_{\mathrm{w}} = 32$ wide beams from tier-1. It then sweeps the child beams in tier-2 that are associated with the best wide beam identified during the first step. The child beam with the highest received power is then selected. In this method, the wide beams are generated using the alternative minimization method with a closed form expression (AMCF) algorithm proposed in [Qi et al., 2020]. For M_{w} wide beams swept in tier-1, the number of narrow beams under each wide beam are $\lceil Q/M_{\mathrm{w}} \rceil$. The total number of beam measurements required is then given by $M_{\mathrm{w}} + \lceil Q/M_{\mathrm{w}} \rceil$.

4. *Binary Hierarchical Beam Search:* The BS performs a binary tree search on the narrow-beam DFT codebook of size Q. The search is organized into $\log_2 Q$ hierarchical layers. At each layer, the BS divides the current angular search space into two partitions and sweeps two corresponding wide beams generated using the AMCF technique. This process continues until a single narrow beam in Q is identified.

5. *Fixed Beam Selection:* Based on the adaptation of the method proposed in [Jalali et al., 2024], the BS selects a fixed subset of RSSI values collected over a subset of narrow beams from the adopted 0-DFT codebook to train a CNN to associate the correct/best-oriented beam between the BS and UE. The

RSSIs over the fixed selected subset of narrow beams are used as the CNN's inputs and the index of the corresponding optimal narrow beam from the 0-DFT codebook as the output. Note that this learning-based approach uses the same number of inputs as the proposed approach, but does not account for the importance of input features (RSSIs) when selecting the input subset for model training.

To evaluate the explainability and robustness of the proposed DkNN-based beam classifier, we compare its credibility estimates with the outputs of a standard softmax classifier, typically interpreted as model's confidence estimates [Papernot and McDaniel, 2018], with both classifiers using the same neural network architecture.

2.6.4 Performance metrics

We consider the following metrics for performance comparison:

Top-k accuracy

This measures the accuracy when the optimal beam index is among the top k predicted beams. Beam alignment is improved by sweeping the top k beams with the highest predicted probabilities, enabling over-the-air refinement with k additional overhead.

Beam Sweep Time (T_{sweep})

This is defined as the time taken to sweep a desired number of beams, and is linearly proportional to the number of beams swept. Typically, IA is performed periodically and constitutes T_{sweep} and the beam prediction time T_{predict}, defined as the time it takes to process the collected RSSI data and predict the beam it belongs to, such that the total IA time $T_{\text{IA}} = T_{\text{sweep}} + T_{\text{predict}}$. While T_{predict} depends on the processing capability at the BS, $T_{\text{sweep}} = N_b t_s$ depends on the number of candidate beams scanned N_b, where $N_b = \widetilde{M}_{\text{w}}$, and t_s is the time required to scan a single candidate

beam [Cousik, 2022]. Generally, $T_{\text{sweep}} \gg T_{\text{predict}}$, and therefore we consider the beam sweep time for comparing different schemes.

Effective Spectral Efficiency

This quantifies the achievable rate of a communication system while accounting for the overhead associated with beam alignment. It is defined as the proportion of channel resources dedicated to data transmission after subtracting the time spent on beam alignment during the channel coherence period. Mathematically, it is expressed as:

$$\text{SE} = \frac{T_{\text{frame}} - T_{\text{IA}}}{T_{\text{frame}}} \log_2 (1 + \text{SNR}) , \tag{2.13}$$

where T_{frame} represents the duration of one frame during which the channel response remains fixed, i.e., the total time for a communication session [Rezaie et al., 2021].

2.6.5 Performance Evaluation

T-SNE Visualization

Fig. 2.3 shows a t-distributed stochastic neighbor embedding (t-SNE) visualization of data points from the target and DT scenario. The goal of t-SNE is to reduce the dimensionality and visualize high-dimensional data in a two-dimensional (2D) plane, preserving the structure and relationships between points. It achieves this by minimizing the Kullback-Leibler (KL) divergence between the joint probability distributions in the original and reduced spaces. The figure reveals a relatively uniform scatter with a central clustering, indicating that the high-dimensional data has been effectively projected into a lower-dimensional space by t-SNE, while preserving local neighborhood structures and relationships. Since the two datasets share the same layout with slight modifications, the data distributions exhibit similarity despite the introduction of minor imperfections in the DT-aided synthetic data.

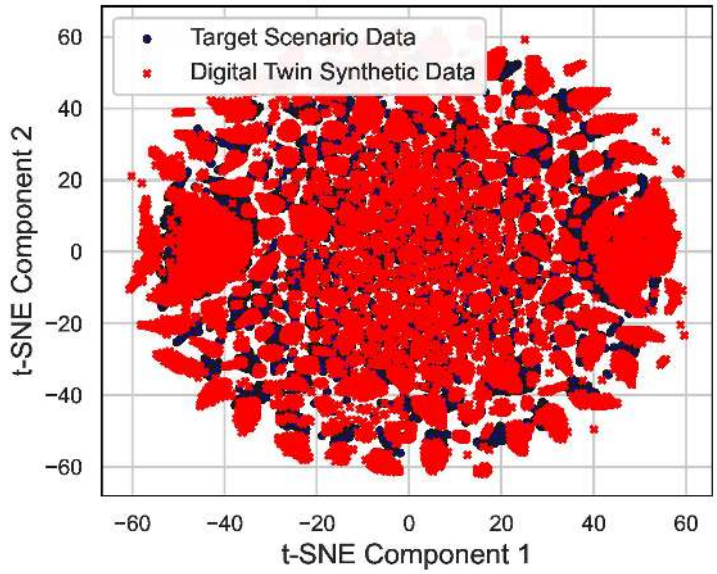


Figure 2.3: T-SNE visualization of the generated data from the target scenario and digital twin scenario.

Analysis of XAI-based Input Feature Selection

We arrange the obtained feature importance scores into rankings as shown in Fig. 2.4. By analyzing these rankings, we can identify the most and the least important features for the DL-based BAE decisions. The horizontal axis shows the average impact on output action, and the vertical axis shows the most influential input features in decreasing order of their magnitude. From Fig. 2.4, it can be observed that the top most essential features selected using Algorithm 1 are mostly in agreement irrespective of the CNN/DNN based model architecture. For visualization purposes, we display the top 12 selected features, while the combined effect of the remaining features is represented as "Others." Further, to select the subset of important features, we set the threshold δ at 71%, 82%, 92%, 96%, and 99%, corresponding to the selection of $\widetilde{M}_{\mathrm{w}} = 2, 4, 8, 12$, and 24 important features, respectively.

Next, we analyze the performance of the DNN models trained on subsets of

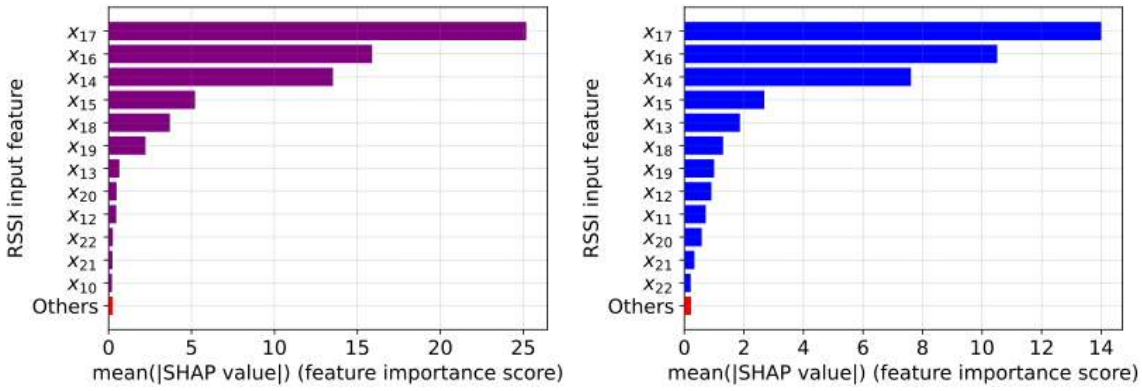


Figure 2.4: Feature importance rankings for the beam classifier generated using CNN (left) and fully connected DNN (right).

importance-based input features, i.e., the RSSIs over selected sensing beams. While similar analyses apply to CNN-based beam prediction, initial experiments revealed that the pre-trained DNN outperforms CNN in *Top-k* accuracy on the site-specific dataset, leading to the selection of the DNN-based BAE for further evaluation. Moreover, we denote the model trained on synthetic/augmented data as the DNN-TL-synthetic and the model purely trained on real-world data as DNN-real, respectively.

Analysis of Model Refinement using Transfer Learning

Fig. 2.5 demonstrates the *top*-2 prediction accuracy versus the importance-based features selected using Algorithm 1. The DNN-TL-synthetic model is initially pre-trained on $57,144$ synthetic data points generated from the DT scenario, where the building positions include a modeling error of 2 meters. In addition, a separate real-world dataset comprising $24,485$ data samples is used to train a dedicated DNN-real model. A subset of the real-world dataset is further utilized to fine-tune the DNN-TL-synthetic model with varying augmentation sizes. For a given number of importance-based selected features, a dedicated DNN-TL-synthetic model is trained, which requires an average training time of approximately 3 minutes per configuration.

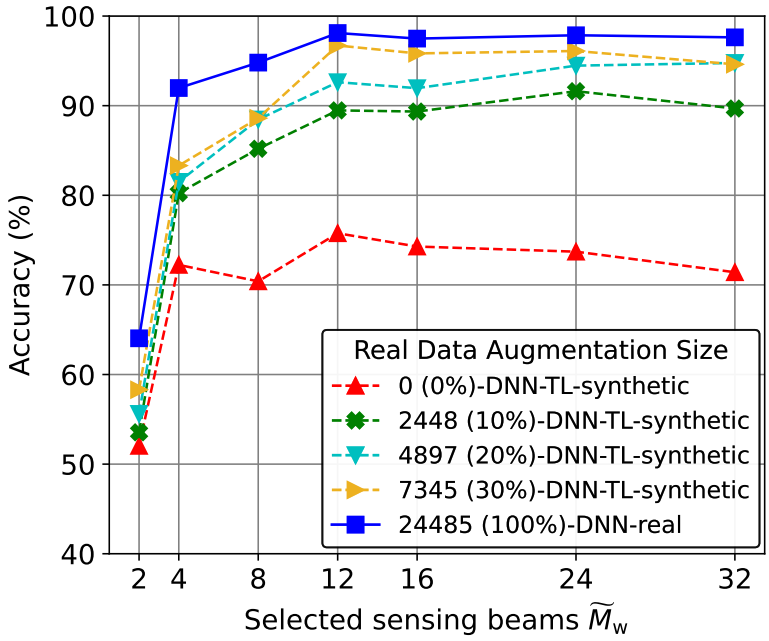


Figure 2.5: Top-2 accuracy vs. number of selected sensing beams/features. The DNN-TL-synthetic model is trained using synthetic data samples from the DT environment and fine-tuned using real-world data augmentation.

Fig. 2.5 illustrates that evaluating the DNN-TL-synthetic model trained purely on synthetic data performs poorly when tested on the unseen target data. By training with limited real data augmentation, e.g, 10% of the destination domain dataset $\mathcal{D}_{\mathrm{R}}$, yields a notable improvement in beam alignment accuracy compared to the scenario where no samples captured at the destination domain are used for model fine-tuning. The figure highlights that with 30% real data augmentation, the DNN-TL-synthetic model, leveraging RSSI values over $\widetilde{M}_{\mathrm{w}} = 12$ importance-based features, attains approximately 98% of the accuracy performance of the DNN-real model trained on the complete real-world dataset. Notably, the DNN-TL-synthetic model achieves this performance while requiring 70% fewer real-world data samples, showcasing model refinement through transfer learning effectively reduces the data collection overhead in real-world systems. For the remaining simulation results, we

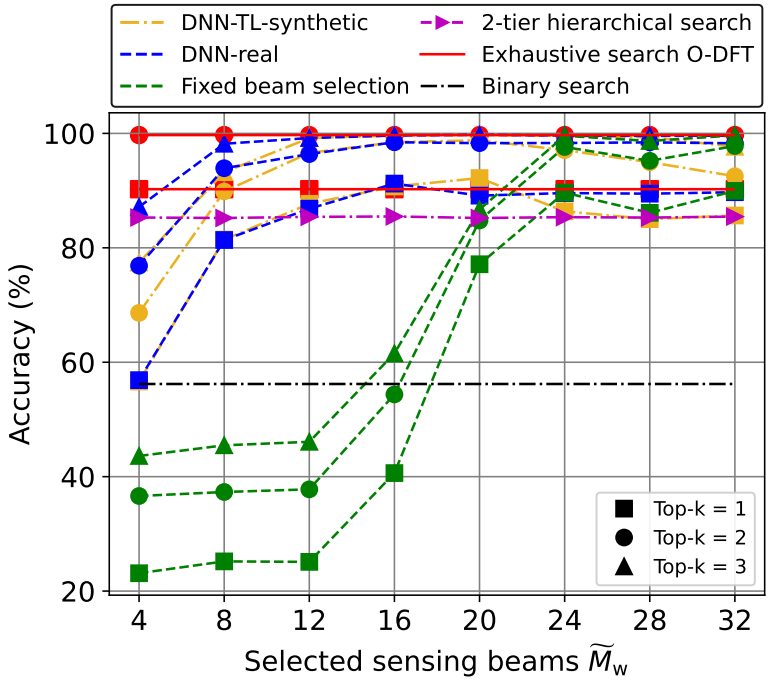


Figure 2.6: Top-k accuracy performance of different schemes under measurement noise.

consider the DNN-TL-synthetic model fine-tuned with only 30% of the real-world data samples, as it efficiently calibrates mismatches between real-world model and digital replica with minimal real-world data requirement.

Analysis of Beam Alignment Accuracy under Measurement Noise

The accuracy of beam prediction depends on the measured power of beamforming signals, where noise in these signals can degrade beam alignment accuracy. The DNN-TL-synthetic model is trained with no measurement noise present in its training data, but it is then exposed to noisy signal measurements during the testing/validation stage. In our analysis, the noise power ranges considered while generating the noisy signal are between -94dBm and -114dBm, and randomly sampled to generate the unseen target data. For a fair comparison, we ensure that all schemes consider the same noise level.

Fig. 2.6 shows the *top-k* $=\{1, 2, 3\}$ accuracy against the number of importance-based features $\widetilde{M}_{\mathrm{w}}$. The exhaustive search achieves the highest accuracy due to its low susceptibility to noise in beam measurements. However, this accuracy comes at the expense of excessive training overhead and occasional search errors caused by the induced noise, resulting in beam alignment mismatches. The binary search performs the worst among the traditional beam sweeping-based baselines since the binary search introduces more layers and is thus more prone to error propagation; a suboptimal decision in any upper layer may lead to an incorrect final beam selection due to noise or imperfect beam patterns. The importance of feature selection becomes evident from the performance of the fixed beam search approach, which treats all features/beams as equally important. It requires sweeping at least 24 sensing beams to achieve an accuracy of 87%. The proposed method can achieve a beam alignment accuracy of 87% using measurements from $\widetilde{M}_{\mathrm{w}} = 12$ importance-based sensing beams and can attain 95% accuracy by searching an additional $k = 2$ narrow beams. With $\widetilde{M}_{\mathrm{w}} = 12$ important features, the fine-tuned model achieves *top-3* accuracy of $\approx 97\%$ outperforming the exhaustive search method. This improvement is mainly attributed to eliminating less important and non-informative RSSI inputs, which would complicate the learning process with noisy measurements.

Analysis of Average SNR

The beam alignment accuracy considers the probability of finding the optimal narrow beam. With a large O-DFT codebook, adjacent narrow beams may have similar beamforming gains. As a result, operators may be more interested in the average SNR achieved after beam alignment.

Fig. 2.7 illustrates the average SNR achieved by the proposed method and the benchmark schemes with the increasing number of selected features $\widetilde{M}_{\mathrm{w}}$. The exhaustive search achieves close-to-optimal average SNR of the SVD-based solution as it searches over all possible beams in the O-DFT codebook. The DNN-TL-synthetic model fine-tuned with 30% real-world data, remarkably matches the average SNR of the DNN-real model trained on the entire real dataset, using just 12 out of 32

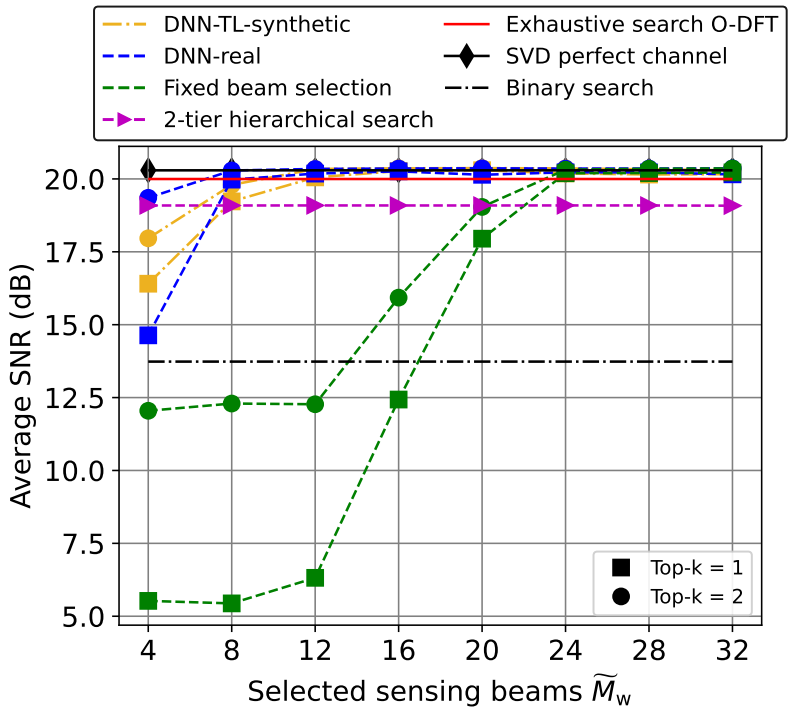


Figure 2.7: Average SNR versus the number of selected sensing beams/features for different schemes.

importance-based sensing beams and outperforms all traditional benchmarks including the exhaustive search. This is due to the fact that the DL methods can also learn the statistical property of noise, which is lacking in conventional methods. Furthermore, the proposed SHAP-based beam selection method reduces the beam sweeping overhead by 50% compared to the fixed beam search approach, which does not account for the relevance of input features in the prediction process. The degraded performance of fixed beam search approach may be attributed to the fact that the best beam may be overlooked since it might not be included in the measured set during training. The superior performance of the proposed DNN-based BAE stems from the SHAP-based feature selection approach, which identifies the "quality of information" in these selected subsets of features to be more valuable and sufficient for the model in identifying the narrow beams best oriented to the UEs. Furthermore, the proposed DNN-TL-synthetic model delivers performance comparable to

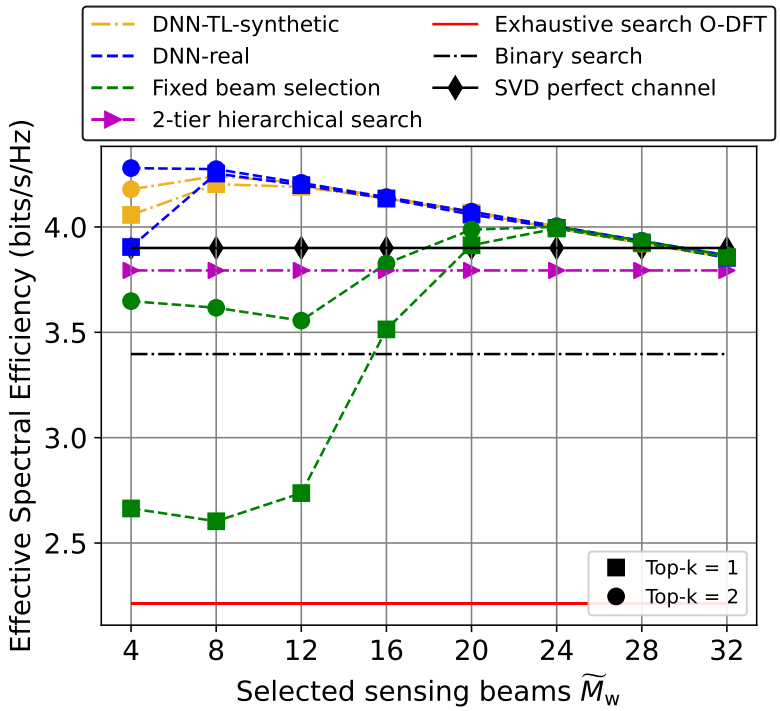


Figure 2.8: Effective spectral efficiency vs. number of selected sensing beams/features for different approaches.

the exhaustive search solution with $\approx 89\%$ reduction in beam training overhead compared to the exhaustive search, demonstrating its effectiveness in challenging NLOS environments.

Analysis of Effective Spectral Efficiency

Fig. 2.8 shows the achievable effective spectral efficiency using the proposed and benchmark schemes across different number of selected features $\widetilde{M}_{\mathrm{w}}$. Contrary to the SVD-based solution and the exhaustive search approach, where the number of input features is fixed at 32 and 128, respectively, the proposed solution adapts the candidate sensing beams $\widetilde{M}_{\mathrm{w}}$ based on the selected features determined by their feature importance. With $\widetilde{M}_{\mathrm{w}} = 8$ beams, the proposed DNN-TL-synthetic model can outperform the SVD-based solution in terms of effective spectral efficiency metric and delivers the same performance as the DNN-real model by searching an additional

$k = 2$ narrow beams. The exhaustive search method requires scanning 128 narrow beams and has the worst spectral effectiveness despite having superior average SNR performance. The hierarchical search and binary tree search based approaches require $M_{\mathrm{w}} + \lceil Q/M_{\mathrm{w}} \rceil$ and $2 + 2\log_2(Q/2)$ beam searches for M_{w} wide beams and Q narrow beams, respectively, and therefore exhibit degraded effective spectral efficiency. An interesting observation is that the effective spectral efficiency has a unique maximizer $\widetilde{M}_{\mathrm{w}}^{*}$: it increases initially with $\widetilde{M}_{\mathrm{w}} \leq \widetilde{M}_{\mathrm{w}}^{*}$ as the beam alignment probability improves with $\widetilde{M}_{\mathrm{w}}$. However, as $\widetilde{M}_{\mathrm{w}}$ increases beyond $\widetilde{M}_{\mathrm{w}}^{*}$, this gain is offset by the increased overhead and reduced time for the data communication.

Computation, Model and Time complexity

The computation complexity of the proposed DL-based BAE is determined by the beam sweeping complexity and feedback complexity, and is linearly proportional to the number of beams swept. With the proposed DL-based BAE , all N_{U} UEs can measure and report the RSSIs over the $\widetilde{M}_{\mathrm{w}}$ sensing beams simultaneously when the BS sweeps the selected sensing beams. If the BS chooses to sweep the top-k predicted beams, the selected beams could vary across different UEs. Hence, the beam sweeping complexity is $\widetilde{M}_{\mathrm{w}} + N_{\mathrm{U}} \times k \cdot \mathbb{1}_{\{k>1\}}$, whereas the feedback complexity will be $N_{\mathrm{U}} \times \widetilde{M}_{\mathrm{w}} + N_{\mathrm{U}} \cdot \mathbb{1}_{\{k>1\}}$, since if the BS chooses to sweep additional beams, each UE only needs to feedback the index of the best beam. The model complexity of the proposed DL-based BAE depends on the structure of implemented DNNs (number of hidden layers and the number of corresponding neurons). The model complexity is primarily dominated by the update process of the network weight parameters. Let H_l denote the number of neurons in hidden layer L. The total number of scalar parameters is $\mathcal{O}\left(|\mathbf{x}_S| \cdot H_1 + \sum_{l=1}^{3} H_l \cdot H_{l+1}\right)$, where $|\mathbf{x}_S| = \widetilde{M}_{\mathrm{w}}$ is the input size, i.e, the number of input neurons. The update process of the loss function is linear to the mini-batch size B since the optimizer scans all mini-batch samples. As a result, an update of all parameters incurs a computational complexity of $\mathcal{O}\left(B \cdot \left(|\mathbf{x}_S| \cdot H_1 + \sum_{l=1}^{3} H_l \cdot H_{l+1}\right)\right)$.

Table 2.3 shows the complexity comparison in terms of the trainable parameters,

Table 2.3: Model and time complexity for proposed beam alignment method

No. of Sensing Beams/Features $\widetilde{M}_{\mathrm{w}}$	**Trainable Parameters**	**Top-1/Top-2/Top-3 Accuracy (%)**	**IA Time (ms)** $T_{\mathrm{IA}} = T_{\mathrm{sweep}} + T_{\mathrm{predict}}$
2	29,184	44 / 55 / 63	0.15625 + 0.0427 = 0.19895
4	29,312	56 / 68 / 77	0.3125 + 0.0428 = 0.3553
8	29,568	81 / 88 / 93	0.625 + 0.0425 = 0.6675
12	29,824	87 / 96 / 97	0.9375 + 0.0435 = 0.9850
16	30,080	90 / 98 / 99	1.25 + 0.0414 = 1.2914
24	30,592	87 / 97 / 99	1.875 + 0.0478 = 1.9228
32	31,104	88 / 93 / 98	2.5 + 0.0416 = 2.5416

accuracy, and IA time for different numbers of selected sensing beams $\widetilde{M}_{\mathrm{w}}$. From the table, it is evident that the beam sweep time dominates the overall IA time. For instance, 5G NR specifies a duration 5ms for sweeping across a total of 64 sensing beams. Based on this, the allowed scanning time per beam is 0.078 ms [Giordani et al., 2018], [Cousik, 2022]. Compared to the exhaustive search method requiring $\approx$ 10ms for sweeping across 128 narrow beams and predicting the optimal beam, the proposed DL-based BAE finishes beam sweeping in $\approx$ 0.98ms, using just 12 sensing beam while maintaining close to exhaustive search accuracy. Overall, using just 12 importance-based features out of full set of 32 features/sensing beams, the proposed DL-based BAE can reduce the trainable parameters by $\approx$ 4.29%, and the beam sweeping time by $\approx$ 90.5% compared to the exhaustive search method, respectively.

The prediction times for proposed DL-aided BAE and the learning and non-learning based schemes are tabulated in Table 2.4. Note that the prediction times depend on the system's computational resources. The computational capabilities of BSs in practical networks typically surpass those of the machines used in our simulations. As a result, the average time required for training and evaluating beam alignment models, the SHAP based feature selection and model robustness assessment can be further reduced in real-world deployments.

The proposed DL model training and the post- hoc SHAP-based feature selection

Table 2.4: PREDICTION TIME (in ms) FOR DIFFERENT ALGORITHMS

No. of Sensing Beams/Features $\widetilde{M}_{\mathrm{w}}$	**DNN-TL-synthetic (proposed)**	**DNN-Real**	**Fixed Beam Selection**
4	0.0428	0.0429	0.0865
8	0.0427	0.0456	0.0500
12	0.0475	0.0484	0.0490
16	0.0414	0.0786	0.0465
20	0.0466	0.0479	0.0460
24	0.0478	0.0432	0.0461
28	0.0745	0.0427	0.0489
32	0.0416	0.0486	0.0507
Non-learning Schemes with Fixed Number of Beam Measurements			
Exhaustive Search (128 beams)	0.0591		
2-tier Hierarchical Search (36 beams)†	1.7110		
Binary Search (14 beams)⋆	0.9519		

† 32 wide beam sweeps + 4 narrow sweeps per wide beam.

⋆ Computed as $2 + 2\log_2(128/2)$ narrow beam sweeps.

is site-specific and conducted offline assisted by the DT. After training, a single simplified model using the selected features is deployed for beam prediction at that site. If the channel statistics change significantly due to model drift or seasonal variations, the retraining of the deployed model and update of feature selection would be necessary. In such scenarios, the DNN may either be trained from scratch or initialized with existing weights and fine-tuned using data from the new environment. Retraining is triggered when beam alignment performance (e.g., Top-k accuracy or L1-RSRP difference) falls below a threshold. Since large-scale environmental properties tend to vary gradually, retraining and feature selection are anticipated to be needed only occasionally.

2.7 Conclusions

In this chapter, we have developed an explainable DL-based BAE for mmWave MIMO systems that uses RSSI measurements from wide sensing beams to predict optimal narrow beams efficiently. By generating synthetic data with a site-specific DT and refining the pre-trained model through transfer learning, we reduce real-world data requirements by up to 70%. We leverage the model-agnosticism and explainability power of SHAP to design an efficient sensing beam selection strategy to further reduce the beam training overhead by prioritizing critical features and cutting beam training overhead by at least 62% while preserving near-optimal spectral efficiency.

Chapter 3

ROBUST MILLIMETER WAVE BEAM ALIGNMENT FOR AI-NATIVE 6G NETWORKS

3.1 Background and Overview

AI-based solutions have recently attracted great interest, enabling learning from data and adapting to dynamic conditions [Xue et al., 2024a]. By incorporating real-world data, such as radar measurements [Demirhan and Alkhateeb, 2022], camera images [Zhang et al., 2024], and global navigation satellite system (GNSS) coordinates [Heng and Andrews, 2021], DL-based systems can predict optimal beams for beam alignment with greater speed and accuracy in mmWave systems. However, while DL-aided approaches have demonstrated significant improvements in beam alignment efficiency [Demirhan and Alkhateeb, 2022, Zhang et al., 2024, Heng and Andrews, 2021], a key challenge is their vulnerability to out-of-distribution and adversarial inputs. DL-based models can be sensitive to biases in the training data, which can lead to poor performance or even discriminatory outcomes when the model is applied to new data that differs from the training data [Hamon et al., 2020]. Ensuring the resilience of beam selection against input perturbation and adversarial attacks is an important requirement of trustworthiness and is essential for standardization and commercial deployment [Khan et al., 2024]. Malicious actors can clandestinely inject erroneous data into a network, thereby causing the beam alignment engine (BAE) to make false predictions with high confidence, significantly degrading performance.

DL-aided models employed as beam classifiers are highly vulnerable to outlier inputs and lack robustness in confident predictions in the inference process. Determining the confidence of DL-based classifiers can be challenging due to the lack of proper calibration in the softmax layer's output probabilities, which often overes-

timate confidence on out-of-distribution inputs, resulting in a significantly reduced beam prediction accuracy in test time. To address this, the Deep k-Nearest Neighbors (DkNN) algorithm [Papernot and McDaniel, 2018] enhances both robustness and interpretability in beam prediction tasks by integrating the k-nearest neighbors approach with the neural network layers. By comparing test inputs with their nearest training data points, DkNN identifies out-of-distribution and adversarial examples, providing a certified defense mechanism with interpretable insights into beam predictions.

This chapter presents a robust and explainable DL-based beam alignment engine (BAE) to predict mmWave beams during the initial access (IA) process with minimal beam sweeping overhead. The proposed solution is a convolutional neural network (CNN)-based beam predictor that utilizes RSSI feedback of a finite set of sensing beams (i.e., DFT codebook) to accurately predict the optimal narrow beams from the O-DFT codebook for IA and data transmission. The developed solution significantly enhances beam training efficiency by eliminating the need for exhaustive searches within the O-DFT codebook. To further add confidence, interpretability, and robustness in beam predictions, we employ the Deep k-Nearest Neighbors (DkNN) algorithm [Papernot and McDaniel, 2018], which evaluates the internal representations of the neural network during test time. This provides a reliable measure of prediction credibility by examining how well test inputs align with the model's training data. Our model-agnostic framework not only enhances interpretability but also effectively identifies out-of-distribution or outlier inputs, improving the system's resilience to adversarial attacks and ensuring robust, reliable performance.

3.2 System Model and Problem Formulation

We consider a downlink mmWave communication system, where the BS features a uniform linear array (ULA) with N_{BS} antenna elements to communicate with N_{U} single-antenna UEs. We concentrate on the scenario of multi-user beamforming, where the BS communicates with very UE using only a single stream. The channel

from the BS to the UE_u can be expressed based on geometric channel modeling as

$$\mathbf{h}_u = \sum_{l=1}^{L} \alpha_{u,l} \mathbf{b}\left(\phi_{u,l}\right), \tag{3.1}$$

where L denotes the number of paths, $\alpha_{u,l}$ represents the complex path gain for the l-th path, $\phi_{u,l}$ is the angle of departure for the l-th path, and $\mathbf{b}\left(\phi_{u,l}\right)$ represents the array response vector, which is given by

$$\mathbf{b}(\phi_{u,l}) = \frac{1}{\sqrt{N_{\text{BS}}}} \left[1, e^{j\frac{2\pi}{\lambda} d \sin(\phi_{u,l})}, \ldots, e^{j(N_{\text{BS}}-1)\frac{2\pi}{\lambda} d \sin(\phi_{u,l})}\right]^T, \tag{3.2}$$

where λ is the signal wavelength, and $d = \frac{\lambda}{2}$ denotes the antenna spacing. To mitigate the hardware cost and power consumption of a fully digital system, we adopt analog-only beamforming where the BS has a single common transmit/receive radio frequency (RF) chain shared by N_{BS} antennas. Hence, the beamforming vector designed for the BS is given by

$$\mathbf{w} = \frac{1}{\sqrt{N_{\text{BS}}}} \left[e^{j\varphi_1}, \ldots, e^{j\varphi_i}, \ldots, e^{j\varphi_{N_{\text{BS}}}}\right]^\top \in \mathbb{C}^{N_{\text{BS}} \times 1}, \tag{3.3}$$

where φ_i is the phase shift of i-th antenna element. We assume that the BS adopt a beamforming codebook $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_Q\}$ incorporating Q pre-defined beamforming vectors. The vectors in $\mathbf{W}$ satisfy $\|\mathbf{w}_q\|^2 = 1, \forall q \in \{1, \ldots, Q\}$ to accommodate the constant-modulus constraint of analog beamforming architecture.

During the beam sweeping process, the BS periodically transmits symbols $s_w \in \mathbb{C}$ to the UEs through the beams defined by the matrix $\mathbf{W} \in \mathbb{C}^{N_{\text{BS}} \times Q}$. Following the beam sweeping process, the complex received signal at the UE_u, using the q-th beamforming vector, can be expressed as

$$r_{\text{RSSI},u} = \sqrt{P_{\text{BS}}} \mathbf{h}_u^H \mathbf{w}_q s_w + z_{\text{RSSI},u}, \tag{3.4}$$

where P_{BS} denotes the BS transmit power, $\mathbf{h}_u \in \mathbb{C}^{N_{\text{BS}} \times 1}$ is the channel vector, and $z_{\text{RSSI},u}$ represents the additive complex noise with power σ_z^2. Then, with unit-power

transmitted symbols, the signal-to-noise ratio (SNR) at the UE_u can be expressed as

$$\text{SNR}_u = \frac{P_{\text{BS}} \left|\mathbf{h}_u^H \mathbf{w}_q\right|^2}{\sigma_z^2}. \tag{3.5}$$

For a given BS-UE pair, the optimal beam index q_u^* can be identified by selecting the beam that maximizes the SNR:

$$q_u^* = \underset{q_u \in \{1,2,\ldots,Q\}}{\arg\max} \left(\frac{\left|\mathbf{h}_u^H \mathbf{w}_q\right|^2 P_{\text{BS}}}{\sigma_z^2} \right) = \underset{q_u \in \{1,2,\ldots,Q\}}{\arg\max} \left(\left|\mathbf{h}_u^H \mathbf{w}_q\right|^2 \right). \tag{3.6}$$

Identifying q_u^* through exhaustive beam search over Q codewords results in significant beam training overhead, which will be addressed by the proposed solution next.

3.3 DL-based Beam Alignment

This section describes the details of the DL-based beam alignment method and the proposed DkNN-based robust beam selection framework.

To address the beam training overhead caused by exhaustive search and resulting feedback communication overhead, the proposed solution leverages the RSSI values over a set of compact wide sensing beams M_{w} (i.e., beams from DFT codebook), to select the best narrow beam from the O-DFT codebook, thus, avoiding exhaustive search over the O-DFT codebook. For beam-sweeping, the BS transmits the pilot signals over a smaller set of beams, each at a separate time slot. All UEs connected to the BS measure and report the RSSI values of the M_{w} beams. It is assumed that beam sweeping, measurement, and reporting occur within the coherence time during which the channel remains constant. The reported beam sweeping results for UE u can be written as

$$\mathbf{x}_u = \left[\left| [r_{\text{RSSI},u}]_1 \right|^2 \cdots \left| [r_{\text{RSSI},u}]_{M_{\text{w}}} \right|^2 \right]^T, \tag{3.7}$$

where $[r_{\text{RSSI},u}]_i = \sqrt{P_{\text{BS}}} \mathbf{h}_u^H \mathbf{w}_i s_w + z_i$ is the received signal using the i-th sensing beam, $\forall i \in M_{\text{w}}$.

In the DL-based approach, this beam selection problem is formulated as a classification task. Particularly, the reported beam sweeping results from (3.7) are fed as input to the deep neural network (DNN)-based beam classifier denoted by $f(.;\boldsymbol{\theta}) : \mathcal{X} \rightarrow \mathbb{R}^{Q}$, where $\mathcal{X} \in \mathbb{R}^{M_w}$ is the set of DNN's input corresponding to the RSSI values over M_{w} beams and $\boldsymbol{\theta}$ denotes the DNN's weight parameters vector. The beam classifier outputs the posterior probability distribution $\widehat{\mathbf{q}} = f_q(\mathbf{x}, \boldsymbol{\theta})$ of each narrow beam in Q being the optimal beam. The true beam index is represented as a one-hot vector $\mathbf{q}^{\star} \in \mathbb{R}^{Q}$, and the loss function is formulated as the cross-entropy between the predicted and true distributions:

$$\mathcal{L}(\mathbf{x}_u, \mathbf{q}^{\star}, \boldsymbol{\theta}) = -\sum_{i=1}^{Q} q_i^{\star} \log \widehat{q}_i, \tag{3.8}$$

where i is the index representing each possible beam, $q_i^{\star}$ and $\widehat{q}_i$ denote the true one-hot encoded vector and the predicted probability for beam $i \in Q$, respectively.

While the aforementioned DL-aided approach reduces beam alignment overhead, it still faces issues with robustness and reliability in confidence estimates. DNN models often struggle to quantify prediction confidence due to their black-box nature, making it difficult to trust their decisions in real-world scenarios, where inaccurate beam selection can significantly degrade performance. Next, we propose a framework analyzing the internal representations of the neural networks to strengthen the explainability and robustness of its beam predictions.

3.3.1 Proposed Robust Beam Classifier Framework

To quantify the confidence in the beam prediction task, we utilize the DkNN algorithm from [Papernot and McDaniel, 2018], which incorporates concepts from conformal prediction to analyze the internal representations generated by the DNN/CNN during testing. This method identifies inconsistencies with training observations, thus providing a more reliable measure of confidence than the softmax-based prediction typically used to estimate the model's confidence. The DkNN algorithm is model-agnostic and can be applied to any pre-trained DL model with manageable computational complexity.

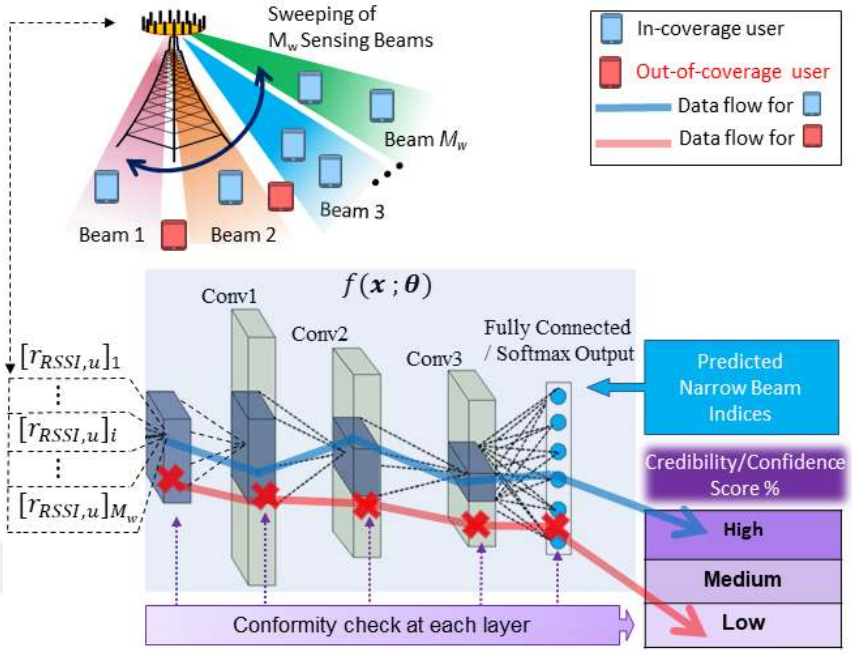


Figure 3.1: DkNN-based credibility assessment of the proposed beam alignment framework.

Let $f(\mathbf{x}_u;\boldsymbol{\theta})$ be a pre-trained DNN-based beam classifier composed of L layers, where each layer is indexed by η, with $\eta \in \{0,\ldots,l-1\}$ as shown in Figure 3.1. After training the DNN for beam selection, the algorithm records the output of each layer f_η for every training point, thus obtaining a per-layer representation of the training data paired with their respective labels. This per-layer representation is used to build a nearest neighbor classifier in the space defined by each layer l to create a representation of the training data at each layer. To efficiently identify nearest neighbors in the high-dimensional spaces produced by these layers, we use locality-sensitive hashing (LSH) [Andoni et al., 2015], which finds similar representations based on cosine similarity. For a given test input $\hat{\mathbf{x}}_u$, we pass it through the DNN to obtain the layer outputs $f_\eta(\hat{\mathbf{x}}_u)$, then apply LSH to find the k nearest neighbors from the training data for each layer's representation. The labels associated with these nearest neighbors are collected for each layer, and these multi-sets of labels are used to compute the final beam selection prediction through a non-conformity

check.

For a given test input $\hat{\mathbf{x}}_u$, the non-conformity of a prediction is defined as the number of nearest neighboring representations found in training data whose label is different from the candidate label $j \in \{1, 2, \ldots, Q\}$, mathematically expressed as:

$$\varrho(\hat{\mathbf{x}}_u, j) = \sum_{\eta \in 1,\ldots,l} \left|\{i \in \Omega_\eta : i \neq j\}\right|, \tag{3.9}$$

where Ω_η is the multi-set of labels for the training points whose representations are closest to the test input's at layer η, and the operator $|\cdot|$ denotes the cardinality of a set, which is the number of elements contained within it.

Before making beam predictions, we compute the nonconformity of a labeled calibration dataset (X^c, Q^c), sampled from the same distribution as the training data (X, Q) but not used for training. The nonconformity values are defined as $\mathcal{C} = \{\varrho(\hat{\mathbf{x}}_u, q) : (\hat{\mathbf{x}}_u, q) \in (X^c, Q^c)\}$, and are then compared to the test input's non-conformity score $\varrho(\hat{\mathbf{x}}_u, j)$ for each candidate beam label j. For input $\hat{\mathbf{x}}_u$ with label j, we calculate the empirical p-value, which represents the fraction of calibration nonconformity scores larger than the test input's score, as:

$$p_j(\hat{\mathbf{x}}_u) = \frac{|\{\varrho \in \mathcal{C} : \varrho \geq \varrho(\hat{\mathbf{x}}_u, j)\}|}{|\mathcal{C}|}. \tag{3.10}$$

The predicted beam label is the one with the highest p-value, as given by

$$\text{Prediction} = \arg \max_{j \in \{1,2,\ldots,Q\}} p_j(\hat{\mathbf{x}}_u). \tag{3.11}$$

Confidence is defined as one minus the second-highest p-value, i.e., the probability that any label other than the prediction is true, as given by

$$\text{Confidence} = 1 - \max_{j \in \{1,2,\ldots,Q\}, j \neq \text{prediction}} p_j(\hat{\mathbf{x}}_u). \tag{3.12}$$

The prediction credibility is the p-value of the predicted beam label, which measures the degree to which the test input conforms to the training data (equation 3.13).

$$\text{Credibility} = \max_{j \in \{1,2,\ldots,Q\}} p_j(\hat{\mathbf{x}}_u). \tag{3.13}$$

The proposed DkNN-based beam classification algorithm is summarized in Algorithm 2. To evaluate the model's resilience against adversarial/outlier inputs, we

Algorithm 2: DkNN-based credible beam selection

Input: training data (X, Q), calibration data (X^c, Q^c)

Input: trained neural network f with L layers

Input: number $\tilde{k}$ of neighbors

Input: test input $\hat{\mathbf{x}}_u$

for $\eta = 1$ *to* L **do**

  $\Pi \leftarrow \tilde{k}$ points in X closest to $\hat{\mathbf{x}}$ found using LSH tables;

  $\Omega_\eta \leftarrow \{q_i : i \in \Pi\}$

end

$\mathcal{C} = \{\varrho(\mathbf{x}_u, q) : (\mathbf{x}_u, q) \in (X^c, Q^c)\}$;

for $j = 1$ *to* Q **do**

  $\varrho(\hat{\mathbf{x}}_u, j) \leftarrow \sum_{\eta=1}^{L} |\{i \in \Omega_\eta : i \neq j\}|$;

  $p_j(\hat{\mathbf{x}}_u) \leftarrow \frac{|\{\varrho \in \mathcal{C}:\ \varrho \geq \varrho(\hat{\mathbf{x}}_u, j)\}|}{|\mathcal{C}|}$

end

Calculate prediction, confidence, and credibility using equations (3.11), (3.12), and (3.13);

return prediction, confidence, and credibility;

generate the out-of-training data (adversarial) examples by the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014] that aims to manipulate the inputs to a beam classifier by perturbing them in the direction that maximizes the loss function in 3.8 with respect to the true beam labels. To generate the adversarial example, the FGSM computes the perturbations as $\boldsymbol{\delta}_u = \epsilon \operatorname{sign}\left(\nabla_{\mathbf{x}_u} \mathcal{L}(\mathbf{x}_u, \mathbf{q}^\star, \boldsymbol{\theta})\right)$, where ϵ controls the perturbation magnitude under some suitable power constraint with respect to the original RSSI values. The adversarial input computed as $\mathbf{x}_{\text{adv},u} = \mathbf{x}_u + \boldsymbol{\delta}_u$ is used to assess the classifier's robustness to outlier inputs.

To characterize the credibility estimates, we adopt the standard reliability diagrams [Niculescu-Mizil and Caruana, 2005] to visualize the calibration of credibility scores. Reliability diagrams are histograms presenting accuracy as a function of

credibility estimates of the model's prediction. The reliability diagrams bin the classifier's credibility score into S intervals of equal size. A test data point $(\hat{\mathbf{x}}_u, q)$ from the test dataset $(X^{\text{te}}, Q^{\text{te}})$ is placed in bin $\mathcal{B}_s$ if the model's credibility on $\hat{\mathbf{x}}_u$ is contained within the bin, i.e., $(\hat{\mathbf{x}}_u, q) \in \mathcal{B}_s$. For each bin $\mathcal{B}_s$, the within-bin accuracy is defined as:

$$\text{Acc}\left(\mathcal{B}_s\right) = \frac{1}{|\mathcal{B}_s|} \sum_{(\hat{\mathbf{x}}_u, q) \in \mathcal{B}_s} \mathbb{1}_{\{\arg\max_{j \in \{1,2,\ldots,Q\}} p_j(\hat{\mathbf{x}}_u) = q\}}, \tag{3.14}$$

which measures the fraction of test samples within the bin that are correctly classified.

3.4 Simulation Setup

In this section, we provide details of the simulation setup, dataset acquisition, and the DL model architecture.

To simulate the BS-UE mmWave communications, we adopt an urban scenario comprising the downtown sector of Boston with both line-of-sight (LOS) and non-line-of-sight (NLOS) users. The BS employs a ULA with $N_{\text{BS}} = 32$ antennas and is placed at a height of 15 meters while oriented towards the negative y-axis. The BS communicates with single-antenna UEs with a height of 2 meters. The service area measures 200 meters by 230 meters and is discretized into a user grid with a spacing of 0.37 meters. Based on these configurations, a total number of 98,650 downlink UE channels are generated. We construct the channel matrix for every UE position using the DeepMIMO dataset generator [DeepMIMO,] according to the specified parameters and system configuration summarized in Table 3.1. To enhance the stability and efficiency of training, the channel vectors, the channel vectors $\mathbf{h} \in \mathbb{C}^{N_{\text{BS}} \times 1}$ are normalized by the largest absolute value in the channel's matrix. All simulations are performed on a 10-Core Intel(R) Xenon(R) Silver 4114, 2.2GHz system equipped with an Nvidia Quadro P2000 graphics processing unit (GPU).

The BS performs analog beamforming using M_w sensing beam using a N_{BS}-DFT codebook, whereas, for the narrow beam codebook $\mathbf{W}$, we use an O-DFT with

Table 3.1: Simulation parameters.

Name of scenario	**Boston-5G**
Active BS	1
BS transmit power	30 dBm
Active users	900-1622
Number of antennas (x, y, z)	(1, 32, 1)
Carrier frequency	28 GHz
System bandwidth	0.5 GHz
Antenna spacing	0.5
OFDM sub-carriers	512
OFDM sampling factor	1
OFDM limit	1
Number of multi-paths	5

Table 3.2: Architecture and training hyperparameters.

Layer	**Filters**	**Kernel**	**Stride**	**Padding**
Conv 1	32	3×3	1	1
Conv 2	64	3×3	1	1
Conv 3	128	1×1	1	0
Fully connected	128 units			
Activation	ReLU (after each Conv layer)			
Optimizer	Adam, 10^{-3}			
Loss function	Cross entropy			
Epochs	100			

OS factor of 4 to get a total of 128 narrow beams. The parameters of the neural network $\boldsymbol{\theta}$ are learned from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_{u,k}, \mathrm{q}_k^\star) : k = 1, .., K\}$ which is composed of K labeled training samples. Each sample consists of the received power values as the input features and the O-DFT beam index as the target label generated using (3.6). In all experiments, 70% of the data is allocated for training, 10% for validation, and the remaining 20% for testing. The calibration dataset is created by reserving a portion of the test data not utilized for evaluation.

The proposed DkNN beam classifier is implemented using a CNN, owing to its powerful feature extraction capabilities [Krizhevsky et al., 2012]. The designed clas-

sifier has $L = 4$ layers with three convolution layers stacked with a fully connected layer. Each convolution layer is followed by the rectified linear unit (ReLU) activation to provide non-linearity to the convolutional layers. The fully connected layer takes the flattened input vector of local information and outputs the softmax probability distribution. For a quick nearest neighbor search on DkNN, we use an LSH from the FALCONN Python library [Andoni et al., 2015] and employ a grid parameter search to configure the number of neighbors to $\tilde{k} = 10$. The hyperparameters of the DkNN beam classifier are summarized in Table 3.2.

3.5 Performance Evaluation

For comparison purposes, we consider the following benchmark methods: 1) the upper bound beamforming based on the SVD of perfectly known channels under 4-bit quantized phase shifter [Heath et al., 2016]; 2) the DFT-based codebook scanning directions with N_{BS} candidate beams at the BS [Heng and Andrews, 2021]; and 3) the O-DFT codebook [Abdallah et al., 2024] with an oversampling factor $(OS) = 4$ and $N_{\mathrm{BS}} \times 4$ candidate beams at the BS. It is worth mentioning that the SVD upper bound can only be reached when each user's perfect channel is known at the BS. To evaluate the explainability and robustness of the proposed DkNN-based beam classifier, we compare its credibility estimates with the outputs of a standard softmax classifier, typically interpreted as model's confidence estimates[Papernot and McDaniel, 2018], with both classifiers using the same DNN architecture.

To assess the optimal beam alignment accuracy, we use the *top-k* accuracy metric, defined as the proportion of test samples where the optimal beam index falls within the top k predicted beams.

The accuracy of beam prediction is influenced by the measured power of the beamforming signals, where noise in these signals can significantly affect beam alignment performance. We consider two different cases: 1) *DkNN trained without measurement noise* ($\mathrm{DkNN}_{\mathrm{wo}}$) where the CNN model is trained with no noise present in its training data, but it is then exposed to noisy signal during the testing/validation stage; 2) *DkNN trained with measurement noise* ($\mathrm{DkNN}_{\mathrm{wn}}$) where the CNN model

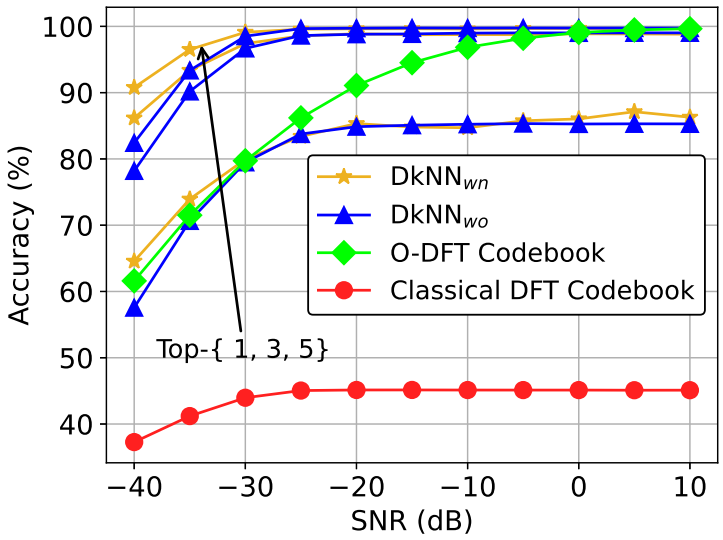


Figure 3.2: Accuracy versus SNR values (dB) for different schemes.

is trained with the expected measurement noise and is then deployed in a network with the expected noise distribution.

3.5.1 Measurement Noise Analysis

Fig. 3.2 compares the accuracy performance of different schemes across different SNR values. In our analysis, the noise power ranges considered while generating the training data are between -28dBm to -90dBm. It is evident that DkNN$_{\text{wn}}$ maintains higher accuracy across expected noise levels compared to DkNN$_{\text{wo}}$. In the NLOS environment, the DKNN trained for expected noise maintains top-5 and top-3 accuracy above 92% until SNR drops below -35 dB, while DkNN$_{\text{wo}}$ sees a more significant decline for higher noise levels. The reason for the improved performance of the DkNN is that it can train the neural network with the received signal containing channel measurement noise and is more robust at low SNR values. In comparison, the O-DFT (x4) codebook-based solution is less robust against noise, with a steady decrease in accuracy below -10dB SNR. On the other hand, the classical DFT codebook solution shows minimal adaptability, remaining below 45% accuracy across all SNR levels, illustrating its limitations in dynamic environments and susceptibility to measurement noise.

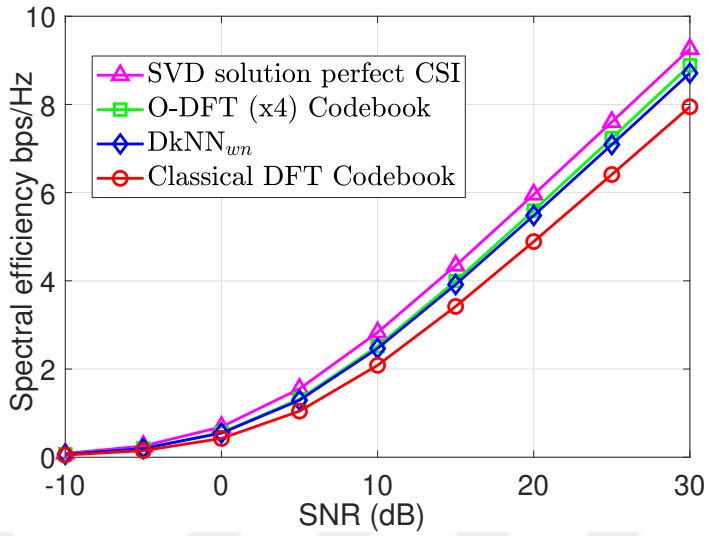


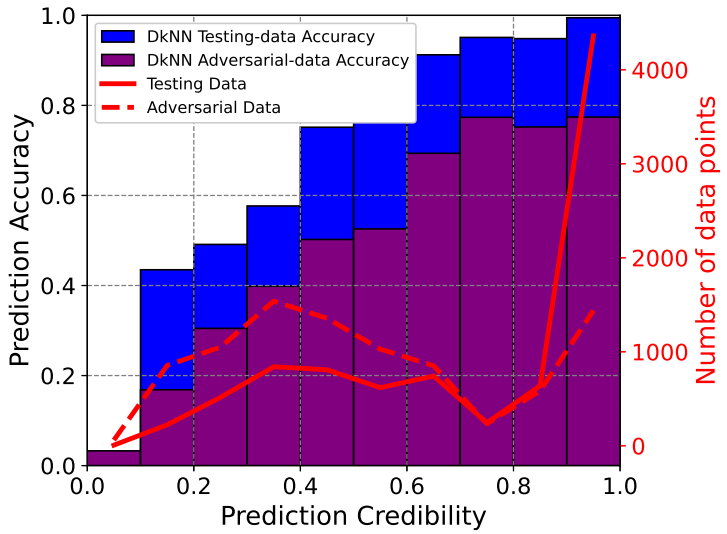
Figure 3.3: Spectral efficiency versus the SNR for different schemes.

Fig. 3.3 illustrates the spectral efficiency versus SNR for various beam alignment methods. The proposed DkNN$_{\text{wn}}$-based algorithm exhibits resilience against measurement noise and achieves approximately 98.5% of the 128-DFT codebook performance in terms of spectral efficiency at all SNR values while reducing the beam training overhead by $\sim$ 75% compared to the O-DFT codebook solution and significantly outperforming classical DFT codebook based solution. Compared to the O-DFT codebook-based solutions, which require an exhaustive beam search over 128 beams, the proposed method requires beam sweeping with only a subset of N_{BS} beams and an additional *top-k* predicted beams.

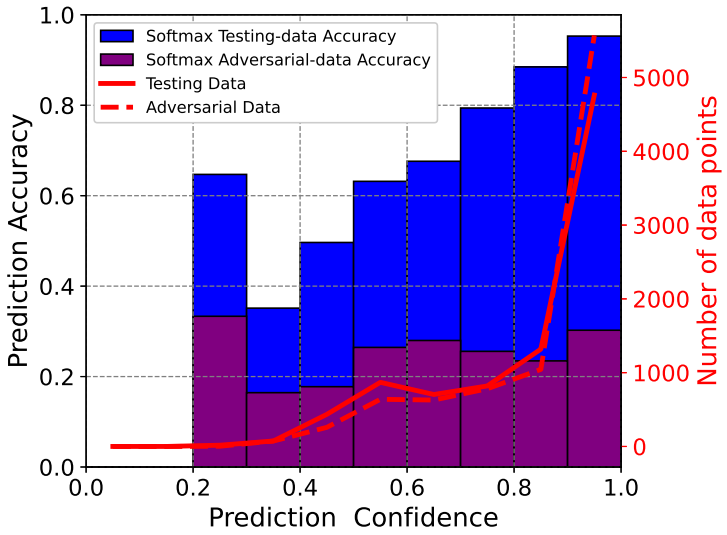
3.5.2 Explainability and Robustness Evaluation

We evaluate the explainability and robustness of the proposed DkNN-based beam classifier by comparing its prediction credibility to that of the softmax classifier. The softmax classifier estimates confidence using output probability distributions but lacks explainability and robustness. In contrast, the DkNN provides interpretability through nearest neighbors, offering human-understandable explanations for intermediate computations at each layer, making it a valuable debugging tool. We show that the softmax classifier is poorly calibrated and overestimates confidence when

predicting out-of-distribution inputs.



(a)



(b)

Figure 3.4: Reliability diagrams for the classifiers on the Boston-5G dataset: (a) Reliability diagram for DkNN Classifier, (b) Reliability diagram for Softmax Classifier.

Figure 3.4 presents the reliability diagrams for the DkNN and the naive softmax classifier. The distribution of credibility/confidence values across the data is given by the number of data points in each credibility bin, reflected by the red line over-

laid on the bars. The softmax classifier lacks calibration as it consistently exhibits high confidence on both test and adversarial data, making it ineffective in identifying outliers. Figure 3.4a demonstrates that the proposed DkNN classifier exhibits superior calibration by assigning low credibility to adversarial samples, effectively filtering outliers. It achieves 5× and 3.5× robustness improvements at credibility thresholds of 0.2 and 0.4, respectively. Figure 3.4b illustrates the softmax classifier reliability diagram, which shows overconfidence, misclassifying more than 80% of adversarial examples with high confidence (>0.9). Note that because of the NLOS environment in the Boston-5G dataset, the test dataset contains many test inputs that are difficult to classify, reflected by the lower mean accuracy of the underlying CNN. Still, the DkNN recovers some accuracy on adversarial examples by leveraging representations from CNN's internal layers and, therefore, is better calibrated than its softmax equivalent: its reliability diagrams follow the linear relation between accuracy and CNN's credibility/confidence.

3.6 Conclusion

In this chapter, we have developed an explainable and robust beam alignment framework for 6G mmWave networks. The proposed solution trains a custom-designed CNN-based BAE by collecting RSSI measurements over a small number of compact wide sensing beams from the DFT codebook to select the best narrow beam from the O-DFT codebook for IA and data transmission. To improve the explainability and robustness of the trained BAE, we have proposed a model-agnostic DL-aided framework utilizing the DkNN algorithm, which inspects the internal representations of the model to evaluate how well their predictions conform with the training data, providing interpretable insights into beam selection and robustness against outlier inputs. Compared to the classical DFT codebook-based solutions, the proposed approach reduces beam training overhead by 75% while achieving near-optimal accuracy. Moreover, the proposed DkNN-based algorithm effectively filters outlier inputs and provides interpretable insights rationalizing the beam prediction decisions, enhancing both explainability and robustness.

Chapter 4

EXPLAINABLE AI-AIDED FEATURE SELECTION AND MODEL REDUCTION FOR DRL-BASED V2X RESOURCE ALLOCATION

4.1 Overview

Cellular vehicle-to-everything (C-V2X) networks have emerged as key enablers for intelligent transportation systems, leveraging existing cellular infrastructure to improve road safety, traffic efficiency, and in-vehicle entertainment. C-V2X technology promises widespread coverage, high reliability, and efficient spectrum use, even in high-mobility scenarios [Bagheri et al., 2021]. Advanced C-V2X applications demand transmission latencies of a few milliseconds and 99.999% reliability [Garcia et al., 2021a]. As vehicular networks expand and radio resource management becomes increasingly complex, existing centralized resource allocation approaches in cellular networks struggle to meet diverse quality-of-service (QoS) requirements, particularly ultra-reliable and low-latency needs.

The recent decade has witnessed a surge in publications unveiling the merits of artificial intelligence (AI) and deep learning (DL) methods, suggesting their potential to mitigate the aforementioned challenges of model-based approaches [Celik and Eltawil, 2024]. In particular, deep reinforcement learning (DRL) has been shown effective in solving hard-to-optimize non-convex combinatorial problems efficiently and has widely been adopted to deal with high-dimensional state-action spaces of vehicular communications applications [Parvini et al., 2024]. Many existing studies have considered different DRL-based solutions for solving resource allocation problems in V2X communication networks [Xiang et al., 2021, Liang et al., 2019, Tian et al., 2022, Kumar et al., 2021]. Progress has also been made to integrate DRL with

federated learning [Li et al., 2022], [Zhang et al., 2020], meta-learning [Yuan et al., 2021], graph theory [Liu et al., 2024] and digital twin-driven V2X networks [Hazarika et al., 2023] for improved performance and fast adaptability of the resource allocation policy in the dynamic V2X environments. Nevertheless, DRL-based resource allocation relies on complex deep neural networks (DNNs) with multiple layers, making them difficult to understand due to their black-box nature and lack of explainability [Khan et al., 2024]. This complexity obscures the influence of input state features on the model's output and the logical reasoning behind decisions. This lack of explainability hinders the adoption of DRL-based resource allocation in safety-critical vehicular applications, as the decision-making processes are not easily understandable.

To strike a good balance between performance and explainability trade-off, explainable AI (XAI) is gaining significant momentum in wireless communications to aid the understanding of complex AI models [Li et al., 2020]. XAI encompasses processes and methods that reveal the inner workings of complex AI models, helping to balance the trade-off between explainability and performance inherent to these models. The Shapley additive explanations (SHAP) method is particularly effective for feature selection, as it assigns importance values to each feature based on their contribution to the model's predictions, allowing for the exclusion of less significant features without sacrificing performance [Guo, 2020]. SHAP has a strong theoretical foundation based on Shapley values in game theory and exhibits several desirable properties compared to other XAI methods. First, SHAP solutions satisfy three essential properties for any additive feature attribution model: local accuracy, missingness, and consistency [Lundberg et al., 2020]. Second, SHAP combines the local interpretable model-agnostic explanations (LIME) [Ribeiro et al., 2016] method, which explains individual predictions using an interpretable model, with Shapley values. This integration allows for faster computation of explanations for complex learning models than directly calculating Shapley values.

XAI-based methods for feature selection have garnered significant attention in wireless communications to determine the primary attributes influencing the model

decision and enhancing model performance. Feature selection aims to identify a minimal set of features that optimizes the objective function without significantly degrading performance. Traditional filter-based feature selection methods assess importance based on data characteristics but overlook algorithm dependence and feature interactions. Wrapper methods (e.g., Recursive Feature Elimination) improve accuracy but are computationally expensive for large datasets. XAI-based feature selection methods are generally architecture-specific, data-dependent, and provide local explanations [Ribeiro et al., 2016]. specific architectures and data types like Grad-CAM [Choi et al., 2020] and LIME [Ribeiro et al., 2016] are limited to specific architectures and data types, provide local explanations, and struggle with instability in complex models. XAI-based feature selection methods such as Grad-CAM [Choi et al., 2020] are useful for deep learning models with convolution neural networks CNN-based architecture. However, they are limited to specific architectures and data types. Similarly, LIME [Guo, 2020], [Ribeiro et al., 2016] offers only local model-agnostic explanations and struggles with instability in complex models due to random perturbations. In contrast, SHAP stands out as a popular and stable XAI method owing to its robust mathematical foundation embedded in the game theory concept of Shapley values. It provides a fair allocation of rewards among inputs, considering their individual/collective contributions and providing both local/global explanations.

The original contributions of this chapter are summarized as follows:

- We formulate the joint optimization problem of spectrum and transmit power allocation with the objective of maximizing the sum rate of vehicle-to-network (V2N) and vehicle-to-vehicle (V2V) links under reliability, latency and transmit power constraints in a URLLC-enabled vehicular network.

- We propose a multi-agent deep reinforcement learning (MADRL) algorithm with centralized learning and decentralized execution to solve the formulated optimization problem. Specifically, multiple DQNs are distributively executed at the V2V transmitters, whereas the weight parameters are centrally trained

at the base station (BS) using a common reward function for all agents to ease implementation and improve stability.

- We propose a novel two-stage systematic explainability framework leveraging feature relevance-oriented XAI to simplify the DRL agents. The former stage involves generating state feature importance ranking of the trained models using the SHAP-based importance scores. The latter stage exploits these importance-based rankings to simplify the state space of the agents, whereby the least important features are removed from the model's input, i.e., by masking feature values and observing the effect on the model's performance using an automated novel variable feature selection method.

- We quantitatively validate the effectiveness of the proposed XAI-based methodology. Via extensive simulations, we demonstrate that our proposed methodology can simplify the model in terms of the average training time, the number of broadcast parameters, and the number of optimal state features required for training for different network configurations without significant performance loss.

4.2 System Model and Problem Formulation

This section first provides the system model of the C-V2X communications network. Then, it formulates a resource allocation problem that aims to maximize the sum-rate for the V2N links as well as to ensure reliable information transmission for the V2V links.

4.2.1 System Model

We consider a single-cell C-V2X communications network covered by a Road-Side Unit (RSU) acting as a single-antenna BS, as shown in Fig. 4.1. Each vehicle is equipped with a single-antenna On-Board Unit (OBU), supporting high data rate uplink services and reliable safety message sharing. C-V2X includes two operation

modes to support diverse vehicular applications: V2N and V2V mode. V2N links support high data rate applications (e.g., video streaming, location tracking, map updates) via cellular interfaces using the Uu interface of 5G new radio (NR) [Garcia et al., 2021a]. V2V links transmit safety-related information through device-to-device (D2D) communications, requiring low latency and high reliability. We represent the sets of V2N and V2V links as $\mathcal{N}$ and $\mathcal{K}$, with cardinalities N and K, respectively. Each V2N link occupies a unique sub-band with fixed transmission power. In cases where $N > K$, we introduce $N - K$ virtual V2V connections. Our focus is on Mode 4 in the cellular V2X architecture, where vehicles autonomously select radio resources for V2V communications [Molina-Masegosa and Gozalvez, 2017]. Each V2V link can reuse an orthogonal sub-band of the V2N links to improve spectrum utilization, with multiple V2V links potentially reusing the same sub-band, which may cause interference.

Channel Models

Consider a slotted communication system with scheduling time slots indexed by t, where each slot has a duration of τ. We consider a quasi-static block fading channel model where orthogonal frequency division multiplexing (OFDM) technology is utilized to group consecutive subcarriers to form a spectrum sub-band. The channel coherence time is given by $T_c = \sqrt{\frac{9}{16\pi f_D^2}}$, where $f_D = \frac{f_c \cdot v_s}{c}$ is the Doppler frequency, f_c is the carrier frequency, v_s is the vehicle velocity, and c is the speed of light [Li et al., 2024]. $T_c > \tau$ holds for moderate vehicular speeds, and the channel state information (CSI) can be regarded as constant throughout the slot duration. Then, the direct channel gain of the k-th V2V link over the n-th sub-band in coherence time slot t is expressed as $g_k^{(t)}[n] = \alpha_k^{(t)} \left| h_k^{(t)}[n] \right|^2$, where $h_k^{(t)}[n] \sim \mathcal{CN}(0, 1)$ is a complex Gaussian variable representing the Rayleigh fading, and $\alpha_k^{(t)}$ captures the large-scale fading effect, including path loss and shadowing, which is typically frequency-independent and changes slowly since it primarily depends on the locations of the transmitter and receiver of the V2V pair. The channel gain between the n-th V2N transmitter and the BS over the n-th sub-band at slot t is defined

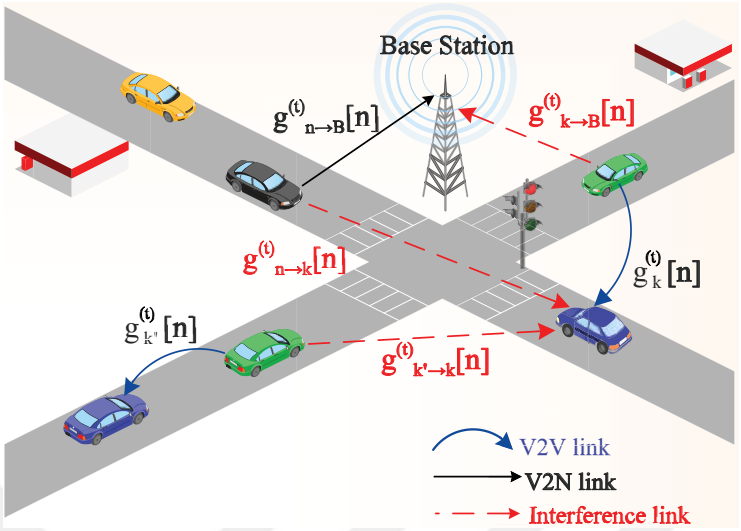


Figure 4.1: System model for the V2X communication network.

as $g_{n\to B}^{(t)}[n] = \alpha_{n\to B}^{(t)} \left| h_{n\to B}^{(t)}[n] \right|^2$, where subscript $n \to B$ denotes the n-th V2N link traversing the BS.

In the case of n-th sub-band's reuse, the interfering channel gains between the i-th and the j-th V2V/V2N links are defined as $g_{i\to j}^{(t)}[n] = \alpha_{i\to j}^{(t)} \left| h_{i\to j}^{(t)}[n] \right|^2$, where $i \neq j$, $i \in \{k, n\}$, $j \in \{k, B\}$, $\alpha_{i\to j}^{(t)}$ represents the large-scale fading, and $h_{i\to j}^{(t)}[n]$ denotes the small-scale fading.

Achievable Rates

The achievable data rate of the n-th V2N link at time slot t is expressed as

$$R_n^{V2N\,(t)}[n] = B_w \log_2 \left(1 + \gamma_n^{V2N\,(t)}[n]\right), \tag{4.1}$$

where B_w indicates the channel bandwidth occupied by each V2N link and $\gamma_n^{V2N\,(t)}[n]$ is the received signal-to-interference-plus-noise ratio (SINR) of the n-th V2N link over the n-th sub-band at time slot t, which is defined as

$$\gamma_n^{V2N\,(t)}[n] = \frac{P_n^{V2N\,(t)} g_{n\to B}^{(t)}[n]}{\sigma^2 + \sum_{k\in\mathcal{K}} \eta_k^{(t)}[n] P_k^{V2V\,(t)}[n] g_{k\to B}^{(t)}[n]}, \tag{4.2}$$

where $P_n^{V2N\,(t)}$ and $P_k^{V2V\,(t)}[n]$ denote the transmit powers of the n-th V2N link and the k-th V2V link over the n-th sub-band, respectively; σ^2 represents the variance of additive white Gaussian noise; and $\eta_k^{(t)}[n] \in \{0,1\}$ is the resource allocation indicator with $\eta_k^{(t)}[n] = 1$ implying the k-th V2V link uses the spectrum of the n-th V2N link and $\eta_k^{(t)}[n] = 0$ otherwise. We assume that each V2V link can only use one sub-band at the same time, i.e., $\sum_{n\in\mathcal{N}} \eta_k^{(t)}[n] = 1$, while the orthogonal sub-band of each V2N link can be reused by multiple V2V pairs.

URLLC-enabled V2V communications are mainly responsible for the reliable dissemination of safety-critical messages using short code block lengths, rendering the classic Shannon capacity no longer appropriate to describe the maximum achievable data rate. In the context of finite block length regime, the maximum achievable information rate of the k-th V2V link at time slot t over the n-th sub-band, which can be decoded with block error probability no greater than ε_k is given by the normal approximation [Popovski et al., 2019]

$$R_k^{V2V\,(t)}[n] = B_w \left(C\left(\gamma_k^{V2V\,(t)}[n]\right) - \sqrt{\frac{V_k^{(t)}[n]}{m}} \frac{f_{Q^{-1}}(\varepsilon_k^{(t)}[n])}{\ln 2} \right), \tag{4.3}$$

where $C\left(\gamma_k^{V2V\,(t)}[n]\right) = log_2\left(1+\gamma_k^{V2V\,(t)}[n]\right)$ is the Shannon rate, $V_k^{(t)}[n] \triangleq 1 - \left(1+\gamma_k^{V2V\,(t)}[n]\right)^{-2}$ is the channel dispersion, m is the blocklength allocated to the k-th V2V link, $\varepsilon_k^{(t)}[n] \in (0,1)$ is the decoding error probability at the k-th V2V receiver over n-th sub-band, $f_{Q^{-1}}(\cdot)$ is the inverse Gaussian tail function $f_Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt$, and $\gamma_k^{V2V\,(t)}[n]$ is the SINR of the k-th V2V link over the n-th sub-band at time slot t defined as

$$\gamma_k^{V2V\,(t)}[n] = \frac{P_k^{V2V\,(t)}[n] g_k^{(t)}[n]}{\sigma^2 + I_k^{(t)}[n]}, \tag{4.4}$$

where $I_k^{(t)}[n] = P_n^{V2N\,(t)} g_{n\to k}^{(t)}[n] + \sum_{k'\neq k} \eta_{k'}^{(t)}[n] P_{k'}^{V2V\,(t)}[n] g_{k'\to k}^{(t)}[n]$ is the collective interference at the k-th V2V receiver from the n-th V2N link and the k'-th V2V links sharing the n-th sub-band. The code block length can be expressed by $m = B_w \Delta_T$, where Δ_T represents the packet transmission latency defined as the time taken for

an entire packet of L bits to be transmitted. In our work, packet transmission latency is fixed to 1 milliseconds to ensure the latency requirement defined in 3GPP TR 36.885 [3GPP, 2023]. Then, for a given payload size of L bits per vehicle, the coding/data rate for the k-th V2V link, $\tilde{r} = \frac{L}{B_w \Delta_T}$. By rearranging Eq. (4.3), the decoding error probability at the k-th V2V receiver can be expressed as

$$\varepsilon_k^{(t)}[n] \approx f_Q \left(\ln 2 \sqrt{\frac{B_w \Delta_T}{V_k^{(t)}[n]}} \left(C\left(\gamma_k^{V2V\,(t)}[n] \right) - \frac{L}{B_w \Delta_T} \right) \right). \tag{4.5}$$

Then, based on the joint coding theory [Wang et al., 2020a], the achievable rate of the k-th V2V link can be expressed as

$$\mathcal{T}_k^{(t)} = \sum_{n=1}^{N} \eta_k^{(t)}[n] \left(R_k^{V2V\,(t)}[n] \left(1 - \varepsilon_k^{(t)}[n] \right) \right). \tag{4.6}$$

4.2.2 Problem Formulation

To maximize the sum-rate for the V2N and V2V links while ensuring reliable information transmission for the V2V links, we jointly optimize the spectrum allocation $\eta_k^{(t)}[n]$ and the V2V transmission power $P_k^{V2V\,(t)}[n]$, $\forall k \in \mathcal{K}$ and $\forall n \in \mathcal{N}$. Specifically, the sum-rate performance metric measures the total data rate achieved by all V2V and V2N communication links in the network. This metric is crucial in V2X communications because it directly reflects the network's ability to efficiently utilize the available spectrum while supporting multiple simultaneous links. A higher sum-rate indicates better spectral efficiency for data-intensive applications and reliable transmission of safety-critical messages. Considering these QoS requirements in vehicular networks, the objective function of our work is to jointly optimize the sum rate for the V2N links and the V2V links. The dynamic multi-objective resource

allocation problem at time slot t is formulated as

$$\underset{\boldsymbol{\eta}^{(t)},\boldsymbol{P}^{(t)}}{\text{maximize}} \left\{ \omega_1 \sum_{n=1}^{N} R_n^{V2N\,(t)} + \omega_2 \sum_{k=1}^{K} \mathcal{T}_k^{(t)} \right\} \tag{4.7a}$$

subject to

$$0 \leq \epsilon_k^{(t)}[n] \leq \varepsilon_{\max,k} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{4.7b}$$

$$0 \leq P_k^{V2V\,(t)}[n] \leq P_{\max}, \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{4.7c}$$

$$\sum_{n \in \mathcal{N}} \eta_k^{(t)}[n] = 1, \quad \forall k \in \mathcal{K} \tag{4.7d}$$

$$\eta_k^{(t)}[n] \in \{0, 1\}. \tag{4.7e}$$

where ω_1, ω_2 represent the weights to balance the sum-rate of V2N and V2V link; and $\boldsymbol{\eta}^{(t)} = \Big[\eta_1[1], \ldots, \eta_k[n], \ldots, \eta_K[N]\Big]$, $\mathbf{P}^{(t)} = \big[P_1^{V2V}[1], \ldots, P_k^{V2V}[n], \ldots, P_K^{V2V}[N]\big]$ are the vectors for the sub-band selection and the power allocations at time slot t, respectively. Then, in (4.7b), $\epsilon_{\max,k}$ is the maximum error probability constraint for the k-th V2V link to ensure a predefined quality of service (QoS). (4.7c) is the maximum power constraint for the k-th V2V transmitter. (4.7d) indicates that each V2V link can only use one sub-band at a time.

The optimization problem (4.2.2) is a non-convex mixed integer nonlinear programming (MINLP) problem and involves sequential decision-making over the time slots. Due to the dynamic nature of the C-V2X networks and the associated short coherence time interval, centralized solutions are inadequate due to difficulties in acquiring the instantaneous CSI of all links [Khan and Coleri, 2022]. To address these issues, we propose to exploit DRL to solve the problem (4.7) in a decentralized fashion. Specifically, the sequential decision-making over time can be encapsulated within a Markov decision process (MDP) framework, and solved using a DRL-based strategy by treating each V2V transmitter as an agent interacting with the unknown environment to maximize the reward. In what follows, we transform the problem of joint spectrum and power allocation into a multi-agent DRL problem.

4.3 Multi-agent Reinforcement Learning-based Solution

In this section, we design a DRL-based algorithm to solve the resource allocation problem in (4.7). To build the foundations for the proposed DRL-based strategy, the description of key components of the RL framework in terms of the state space, action space, and reward function is first provided, followed by the proposed MADRL algorithm.

4.3.1 Problem Transformation into DRL Framework

DRL effectively addresses complex decision-making problems by training an agent to interact with its environment and learn optimal behaviors over time through continuous feedback and adjustment. The agent refines its actions to achieve optimal outcomes within a MDP framework [Mnih et al., 2015]. An MDP is encapsulated by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is the transition probability from states to actions, and $\mathcal{R}$ is the reward for taking action in a state. At each discrete time step t, the agent observes the state $s^{(t)}$, takes action $a^{(t)}$ according to policy $\pi(s^{(t)}, a^{(t)})$, transitions to state $s^{(t+1)}$, and receives a reward $r^{(t)}$. The DRL-based system design aims to learn an optimal policy that maximizes the expected return [Khan and Coleri, 2024]. Therefore, we cast the optimization problem (4.7) as an MDP by mapping the key elements from the DRL framework to the resource allocation problem.

In the multi-agent scenario, each V2V transmitter acts as an agent and interacts with the unknown communication environment. The joint action $\bar{\mathcal{A}}^{(t)} = \left(a_1^{(t)}, \cdots, a_k^{(t)}, \cdots a_K^{(t)}\right)$, with $a_k^{(t)}$ as the k-th V2V agent's action in the time slot t, collectively influences the common environment and directs the agents towards the optimal policy. Hence, in our MADRL framework, different agents compete for limited power and spectrum resources, and the resource-sharing problem is shifted to a fully cooperative one by providing all agents with the same common reward. Next, we introduce the key components of the DRL framework in detail in terms of the state space, action space, and reward function.

State Space

The state of the k-th V2V agent at time slot t, ${s_k}^{(t)}$, includes the channel information and the received interference from other links. Specifically, the channel information for the k-th V2V agent includes the instantaneous channel gain of its own link $g_k^{(t)}[n]$, $\forall n \in \mathcal{N}$, the interference channel gain from the transmitter of the n-th V2N link and the k'-th V2V link, $g_{n\to k}^{(t)}[n]$ and $g_{k'\to k}^{(t)}[n]$, $\forall n \in \mathcal{N}$, $(k' \neq k)$, and the interference channel gain from its transmitter to the BS $g_{k\to B}^{(t)}[n]$, $\forall n \in \mathcal{N}$. The channel information $g_k^{(t)}[n]$, $g_{n\to k}^{(t)}[n]$, and $g_{k'\to k}^{(t)}[n]$, can be accurately estimated by the receiver of the k-th V2V link at the beginning of each scheduling slot, and we assume it is also available instantaneously at the transmitter through delay-free feedback. The CSI for $g_{k\to B}^{(t)}[n]$ is estimated at the BS and broadcast to all vehicles within the BS coverage at the beginning of each scheduling slot [Liang et al., 2019].

Additionally, the aggregate interference from the V2N links and the V2V links over the n-th sub-band in the past time slot $I_k^{(t-1)}[n]$ is included in the state space of the k-th agent, which is expressed as

$$I_k^{(t-1)}[n] = P_n^{V2N(t-1)} g_{n\to k}^{(t)}[n] + \sum_{k'\neq k} \eta_{k'}[n] P_{k'}^{V2V(t-1)}[n] g_{k'\to k}^{(t)}[n],$$

where the first term is the interference from the V2N transmitter and the second term is the aggregate interference from all V2V transmitters using the same n-th sub-band in the previous time slot. It should be noted that the sum interference for the k-th V2V agent is estimated in slot t as follows. At the beginning of the current time slot, the new transmit power and spectrum allocation decision of each V2V agent has not been determined and each V2V agent utilizes the resource decision of the previous time slot to calculate $I_k^{(t-1)}[n]$, although the CSI of the whole network has changed. Then, the current state $s_k^{(t)}$ of the k-th V2V agent is given by

$$s_k^{(t)} = \left\{ \left(g_k^{(t)}[n], g_{k'\to k}^{(t)}[n], g_{k\to B}^{(t)}[n], g_{n\to k}^{(t)}[n],\, I_k^{(t-1)}[n] \right) \mid n \in N \right\}. \tag{4.8}$$

The scale of the state space is $(K+2) \times N$ per V2V agent, which includes N elements for the direct link between the k-th V2V transmitter and its receiver,

$(K \times N - N)$ elements for the interfering channels from other V2V transmitters, N elements for the V2N links, and N elements for the sum interference.

Action Space

The action of each agent at time slot t corresponds to the spectrum sub-band and transmission power selection. While N disjoint sub-bands are preoccupied by the N V2N links and all V2V links share these sub-bands, each V2V agent can only pick at most one spectrum sub-band during the same time slot $\left(\sum_{n\in\mathcal{N}} \eta_k^{(t)}[n] = 1, \forall\mathcal{K}\right)$. The power allocation is discretized in L_Q levels given by $\mathcal{P} = \left\{\frac{P_{\max}}{L_Q}, \frac{2P_{\max}}{L_Q}, \cdots, P_{\max}\right\}$, with $\mathcal{P}$ representing the power action set. Then, the action executed by the k-th V2V agent in the time slot t can be expressed as $a_k^{(t)} = \left[\eta_k^{(t)}[n],\ P_k^{V2V\,(t)}[n]\right]$, which includes $\eta_k^{(t)}[n]$ and $P_k^{V2V\,(t)}[n]$ for spectrum sub-band and power action decision, respectively. Accordingly, each agent has an action vector with the dimension of $L_Q \times N$.

Reward Function

A common reward is designed for all agents to encourage cooperation among the multiple V2V agents. In order to reflect the performance of the decision taken by the agents, the immediate reward function is formulated as

$$\begin{aligned} r^{(t)} =& \lambda_1 \sum_{n=1}^{N} R_n^{(t)\,V2N} + \lambda_2 \sum_{k=1}^{K} \mathcal{T}_k^{(t)} \\ &- \lambda_3 \sum_{k=1}^{K} \left(\max\left(\epsilon_k^{(t)} - \epsilon_{max,k}, 0\right)\right), \end{aligned} \tag{4.9}$$

where the summation terms respectively represent the sum-rate of V2N links, the sum-rate of V2V links, and the reliability requirements of the V2V links; and λ_i, $i \in \{1, 2, 3\}$, are the positive weights to balance the utility and the penalty cost in terms of constraint violations. To align the designed reward function at each step with the network objective (4.7a), the first two terms in (4.9) are included as positive reward elements. Additionally, the reward function incorporates a penalty for failing to satisfy the reliability constraint related to the decoding error probability. The

selection of weight coefficients is critical, as they significantly affect the learning efficiency and convergence of the algorithm. In practical scenarios, these coefficients, denoted as λ_i, are considered hyperparameters and require empirical tuning. Setting equal positive weights for all reward components prioritizes maximizing the sum rate of V2V and V2N links. However, our tuning experience suggests such a design will impede the algorithm's convergence as the agents struggle to learn reliability aspects. This is because the penalty term, based on target error probabilities, is much smaller than other reward components. The reward function components are normalized by setting $\lambda_1 = \frac{1}{5K}$, $\lambda_2 = \frac{1}{R'}$, where $R' = \sum_{k=1}^{K} \sum_{n=1}^{N} C\left(\gamma_k^{V2V\,(t)}[n]\right)$ is the sum V2V Shannon rate and $\lambda_3 = 1$. This hyperparameter selection approach balances the contribution of utility and constraint violation cost while enabling stable updates of the Q-values for improved convergence.

4.3.2 Proposed Multi-Agent DRL-based Algorithm

We propose a MADRL scheme with each V2V transmitter as an agent. The multi-agent scenario breaks the non-stationarity assumption of the MDP as the environment may not be stationary from the agent's perspective as other learning agents update their policies. To effectively handle the non-stationarity issue in multi-agent settings, we propose the centralized training and decentralized implementation approach. Specifically, DQNs are distributively executed at the V2V transmitters, whereas the weight parameters are centrally trained at the BS to ease implementation and improve stability.

The proposed centralized-trained distributively-executed framework is shown in Fig. (4.2). The centralized training and distributed execution procedures are detailed as follows:

Centralized Training

In the training phase, K evaluation DQNs $Q\left(s_k^{(t)}, a_k^{(t)}; \theta_k^{(t)}\right)$ with weight parameters $\theta_k^{(t)}$, and K corresponding target DQNs $Q\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)}\right)$ with weight parameters $\bar{\theta}_k^{(t)}$ are established at the BS, whereas a local DQN $Q^{\,agent}\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta_k}^{(t)\,agent}\right)$, with

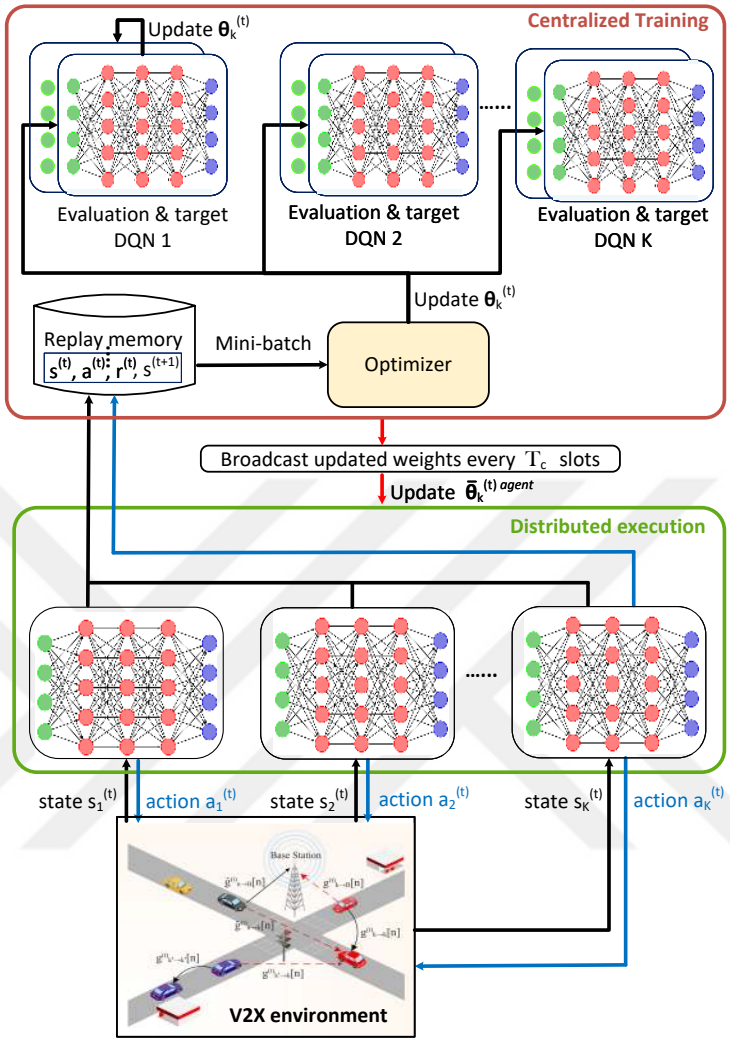


Figure 4.2: Illustration of the proposed multi-agent deep reinforcement learning algorithm.

weight parameters $\bar{\theta}_k^{(t)\,agent}$ is established at the vehicle transmitter as the k-th V2V agent. The local network $Q^{agent}\left(s_k^{(t)}, a_k^{(t)}; \bar{\theta}_k^{(t)\,agent}\right)$ has the same structure as the corresponding evaluation network $Q\left(s_k^{(t)}, a_k^{(t)}; \theta_k^{(t)}\right)$, but is established at the vehicle transmitter instead. The input and the output of each V2V agent are the local state information and the adopted local action (i.e., transmit power and spectrum allocation), respectively. In time slot t, each V2V agent k observes its current state $s_k^{(t)}$ and independently executes the action $a_k^{(t)}$ following the ϵ-greedy policy. The

local observations $\left(s_k^{(t)}, a_k^{(t)}\right)$ of each V2V agent are uploaded to the BS at the beginning of time slot t, whereas the common reward $r^{(t)}$ is evaluated at the BS. Afterwards, the experience $e_k^{(t)} = \left\{s_k^{(t)}, a_k^{(t)}, r_k^{(t)}, s_k^{(t+1)}\right\}$ of each V2V agent k is stored in the replay buffer memory $\mathcal{D}$. Note that all V2V agents share the same reward in the system such that cooperative behavior among them is encouraged, and the selected experiences may include the experiences of different agents at different time slots since a single replay buffer memory is used to store experiences for all agents. In this manner, the BS trains the evaluation DQNs by using the experiences gathered from all agents. Then, by sampling a mini-batch of D experiences from $\mathcal{D}$ in every T_b time slot, the trainer optimizes the weights for the DQNs by minimizing the loss function using the gradient descent technique:

$$\mathcal{L} = \frac{1}{\|D\|} \sum_{e \in D} \left(y^{(t)} - Q(s^{(t)}, a^{(t)}; \theta^{(t)})\right)^2, \tag{4.10}$$

where $y^{(t)} = r^{(t+1)} + \zeta \max_{a^{(t+1)}} Q\left(s^{(t+1)}, a^{(t+1)}; \bar{\theta}^{(t)}\right)$ is the target Q-value generated by the target network. In every T_c time slots, the BS updates the target DQNs with the weights of the evaluation DQNs and broadcasts the latest trained weight parameters ${\theta_k}^{(t)}$ of the evaluation DQN to local agent to update its weights ${\bar{\theta}_k}^{(t)\,agent}, \forall k \in K$. The complete training procedure is summarized in Algorithm 3, which takes the initial DQN model for each V2V agent, batch sampling period $\mathcal{T}_b$, weight copy period $\mathcal{T}_c$, and the number of train episodes $\mathcal{E}_{train}$ as the input and outputs the real-valued trained network's weight vector $\boldsymbol{\theta} = \left\{\theta_k\right\}_{k=1}^{K}$ along with the background dataset $\mathcal{X}_{BG}$ to be used in Section 4.4 for explainability purpose. Algorithm 3 achieves convergence upon completion of the training process.

Distributed Execution

During the distributed implementation phase, the ϵ-greedy algorithm is terminated, i.e., agents stop exploring the environment. Each V2V agent observes its environment state $s_k^{(t)}$ and executes action $a_k^{(t)}$ determined by the maximum action value according to its trained DQN, i.e., $a_k^{(t)} = \arg\max_{a_k^{(t)}} Q^{agent}\left(s_k^{(t)}, a_k^{(t)}; {\bar{\theta}_k}^{(t)\,agent}\right)$. Consequently, all V2V links start transmission with the power level and spectrum

Algorithm 3: Centralized Training Stage

Input: Initial DQN models, $\mathcal{T}_b$, $\mathcal{T}_c$, $\mathcal{E}_{train}$

Output: Trained weights $\boldsymbol{\theta}$, background dataset $\mathcal{X}_{BG}$

Initialize replay memory $\mathcal{D} \leftarrow \emptyset$, and $\mathcal{X}_{BG} \leftarrow \emptyset$;

Random initialize the weights $\theta_k^{(t)}$ of DQNs, and the weights of target DQNs as $\bar{\theta}_k^{(t)} = \theta_k^{(t)}, \forall k \in \mathcal{K}$;

for $e = 0 : \mathcal{E}_{train}$ **do**

  Generate the V2N and V2V communications links;

  for $t = 0 : \mathcal{T}_{steps}$ **do**

    for $k = 0 : K$ **do**

      Observe the state $s_k^{(t)}$;

      Choose the joint action $a_k^{(t)}$ following the ϵ-greedy policy based on $s_k^{(t)}$;

      Evaluate the reward $r^{(t+1)}$ in (4.9) and move to next state $s_k^{(t+1)}$ by executing action $a_k^{(t)}$;

      $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{ s_k^{(t)}, a_k^{(t)}, r_k^{(t)}, s_k^{(t+1)} \right\}$;

      $\mathcal{X}_{BG} \leftarrow \mathcal{X}_{BG} \cup \left\{ s_k^{(t)}, a_k^{(t)} \right\}$;

      if $t > T_b$ **then**

        Randomly sample a mini-batch of experiences from $\mathcal{D}$;

        Update weights ${\theta_k}^{(t)}$ by minimizing the corresponding loss function in (4.10) via the gradient descent technique;

        if $t \,\%\, T_c = 0$ **then**

          Copy weights ${\theta_k}^{(t)}$ of training DQN to target DQN ${\bar{\theta}_k}^{(t)}$;

          Update weights ${\bar{\theta}_k}^{(t)\,agent}$ of local agent with latest weights ${\theta_k}^{(t)}$ of training DQN;

        end

      end

    end

  end

end

return $\boldsymbol{\theta}$, $\mathcal{X}_{BG}$

sub-band determined by their selected actions. Further, the distributed execution phase outputs a hold-out dataset $\mathcal{X}$ containing the environment states and a copy of the action values (Q-values) as the labels. Note that the predicted action of a discrete action space is the index of the *argmax* function, and the experience replay already contains a copy of the action predicted by the trained network. We relegate the detailed design and use of datasets $\mathcal{X}$ and $\mathcal{X}_{BG}$ to Section 4.4.

4.4 XAI Guided Feature Selection and Model Simplification

In this section, we introduce our XAI-based approach to simplify the trained DRL models' complexity by quantifying the relevance of input state features to the output actions for our proposed DRL-based V2X communication system. A novel post-hoc and model-agnostic framework is proposed to explain and reduce the complexity of the trained networks by eliminating some of the input variables using the state feature importance ranking.

4.4.1 Proposed Systematic XAI Methodology

The systematic XAI methodology offers two key benefits: First, it simplifies the input size by eliminating non-influential features through an iterative SHAP-based feature selection algorithm. Second, it reduces the DNN model's complexity by adjusting the architecture based on the refined input size, thereby lowering computational load and minimizing parameter updates for the distributed implementation. Our methodology can be divided into two stages. The first stage involves generating state feature importance ranking of the trained models using the Deep-SHAP explainer. The second stage uses importance-based rankings to simplify the state space of the agents using a novel variable feature selection method.

For notation consistency, we denote the k-th trained V2V agent as the prediction model, i.e., $f_k(\mathbf{x}) \equiv Q_k^{agent}(\mathbf{x})$, which takes real-valued state feature vector $\mathbf{x} = [x_1, \ldots, x_L]$ as an input, where $L = (K+2) \times N$ is the number of input neurons. The prediction model $f_k(\mathbf{x})$ outputs the Q-value of each possible action denoted by $\mathbf{q}_k = [q_{k,1}, \ldots, q_{k,m}, \ldots, q_{k,M}]$, where $M = L_Q \times N$ is the number of output

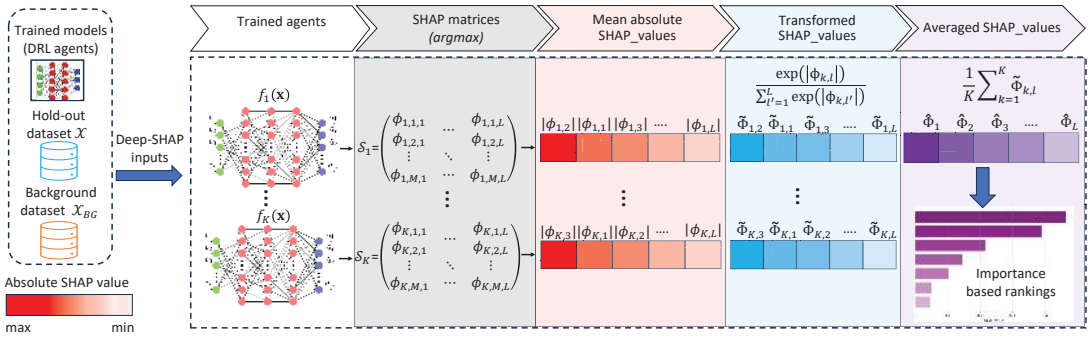


Figure 4.3: The process of generating SHAP_values for multiple trained models using Deep-SHAP.

layer neurons and equal to the cardinality of the action set described in Section 4.3.1. The trained model uses *argmax* operator to select the action maximizing the model's output.

In the first stage of generating state feature importance ranking using the trained agents, we instantiate a DeepExplainer from the SHAP library [Lundberg et al., 2020] and pass the trained agent's DQNs along with the background dataset to it. The background dataset $\mathcal{X}_{BG}$ is generated during the training process and serves as a prior expectation for the sample instances to be explained. To compute the SHAP_values, a hold-out dataset $\mathcal{X}$ is used, which contains the state features and a copy of all possible actions, i.e., Q-values. This dataset is generated by evaluating the performance of the well-trained agents over a set of test episodes. Deep-SHAP produces local explanations in the form of importance score or "SHAP_values," $\boldsymbol{\phi}_k(\mathbf{x}) = [\phi_{k,1}, \ldots, \phi_{k,L}]$, where $\phi_{k,l}$ is the SHAP_value associated with feature $x_l \in \mathbf{x}$ and probable action $q_{k,m}$ of the k-th prediction model. To obtain global explanations, SHAP_values are obtained by repeating this across the entire database, resulting in a SHAP matrix for each sample instance in $\mathcal{X}$. The process for calculating the SHAP_values for the prediction model $f_k(\mathbf{x})$ using Deep-SHAP is delineated in Fig. 4.3, wherein Deep-SHAP first computes the SHAP matrix $\mathcal{S}_k$ for state sample $\mathbf{x} \in \mathcal{X}$ with one row for each action value $q_{k,m}$ and one column per state feature $x_l \in \mathbf{x}$. Then, the SHAP_values corresponding to the maximizing action index are

selected from $\mathcal{S}_k$. Afterward, the absolute mean of the feature column of $\mathcal{S}_k$ is calculated across all data instances in $\mathcal{X}$. The resulting vector of mean absolute SHAP_values $|\boldsymbol{\phi}_k(\mathbf{x})| = [|\phi_{k,1}|, \ldots, |\phi_{k,L}|]$ is sorted in descending order. The first position of the resulting vector contains the most important feature, the second position contains the second most important, and so on. To reveal the meaning behind the SHAP_values in terms of their contribution to the models' outputs, we transform the SHAP_values into a probabilistic distribution $\tilde{\boldsymbol{\Phi}}_k \in [0,1]^L$ through the following softmax transformation:

$$\tilde{\boldsymbol{\Phi}}_{k,l} = \frac{\exp\{|\phi_{k,l}|\}}{\sum_{l'=1}^{L} \exp\{|\phi_{k,l'}|\}}, \forall k \in \mathcal{K}, l = 1, \ldots, L, \tag{4.11}$$

where $\tilde{\boldsymbol{\Phi}}_{k,l} = \left[\tilde{\Phi}_{k,1}, \ldots, \tilde{\Phi}_{k,L}\right]$ denotes the real-valued vector of transformed SHAP_values for the k-th prediction model.

To obtain global state feature importance ranking, the transformed absolute SHAP_values are obtained for each trained agent using the above procedure and averaged across the K trained models. The averaged transformed SHAP_values, denoted by $\widehat{\boldsymbol{\Phi}} = \left[\widehat{\Phi}_1, \ldots, \widehat{\Phi}_L\right]$ are utilized to rank the input state features, which can be interpreted as global state-feature importance scores. Then, by utilizing an appropriate feature selection strategy, the most important state elements contributing to the model outcome are selected, and the model is retrained using the subset of these important states. Note that the presented systematic feature selection strategy in Fig. 4.3 is model-agnostic and applicable to any other type of pre-trained DRL-based radio resource management system.

4.4.2 Post-hoc SHAP-based Input State Selection Algorithm

We devise an iterative state feature selection algorithm based on SHAP-based state feature importance rankings. The post-hoc SHAP-based state feature selection strategy is summarized in Algorithm 4, whose inputs are the K trained agents, hold-out dataset $\mathcal{X}$, background dataset $\mathcal{X}_{\text{BG}}$, and the precision threshold Δ, whereas the outputs are the optimal subset of features retained using the proposed feature selection strategy. The developed variable feature selection strategy selects the number

of important state features by repeatedly evaluating the trained models based on a predefined metric while masking step-by-step the values of the p least important features according to the given importance-based feature ranking. Significant changes in the evaluation results indicate that a sufficiently important state feature has been randomized in the current step. Consequently, all state features randomized in the previous step can be confidently removed from the trained network's input.

While the network is trained to maximize cumulative reward in (4.9), this reward is not the most illustrative way to evaluate its post-hoc performance due to its high variance. To evaluate the trained models' performance, we consider the following reward value averaged across the hold-out dataset

$$\alpha = \frac{1}{\|\mathcal{X}\|} \sum_{i=1}^{\|\mathcal{X}\|} r^{(i)}, \tag{4.12}$$

where $r^{(i)}$ is calculated using the i-th state-action tuple in the hold-out data set, $\mathcal{X}$, as per (4.9).

Algorithm 4 starts with a for loop (Lines $1-4$) that iteratively computes the SHAP matrix $\mathcal{S}_k$, $\forall k \in K$, and obtains transformed SHAP_values, $\tilde{\mathbf{\Phi}}_{k,l}$, $\forall k \in K$, as per (4.11). Then, $\tilde{\mathbf{\Phi}}_{k,l}$, $\forall k \in K$, are averaged across the trained models and utilized to rank the input state features (Lines $5-6$). The generated feature rankings are used to select a subset of state features using the average network performance metric in (4.12). The algorithm first evaluates average network performance in (4.12) with no state features removed, denoted by $\alpha_{original}$ (Line 7). Then, the p least important features are masked according to SHAP importance-based feature ranking, i.e., their values are changed to the respective column variance (Line 10). The trained models are re-evaluated on the modified dataset, and the evaluation results are denoted as $\alpha_{simplified}$ (Line 11). If there is a significant change in the evaluation results, i.e., $|\alpha_{original} - \alpha_{simplified}| \geq \Delta$, where Δ is the precision threshold, then the $p-1$ least important state features are removed and only $L-p+1$ influential state features are retained (Lines $12-15$). Otherwise, the value of p is incremented, and the state feature masking and re-evaluation steps previously defined are performed iteratively (Lines $17-19$). It should be noted that the value of Δ is set to be between 1-10% of

Algorithm 4: Iterative SHAP-based Feature Selection

Input: $f_{k\,(\forall k \in K)}$, $\mathcal{X}$, $\mathcal{X}_{BG}$, Δ

Output: Subset of input state features

for $k = 0, 1, \ldots, K$ **do**
  Generate SHAP matrix $\mathcal{S}_k$ for $f_k(\mathbf{x})$ using $\mathcal{X}$ and $\mathcal{X}_{BG}$;
  Obtain transformed SHAP_values $\tilde{\mathbf{\Phi}}_{k,l}$ using (4.11);
end
$\widehat{\mathbf{\Phi}} \leftarrow$ Average $\tilde{\mathbf{\Phi}}_{k,l}$ across the K agents;
Generate global state-feature importance rankings using $\widehat{\mathbf{\Phi}}$;
$\alpha_{original} \leftarrow$ Evaluate the average network performance on $\mathcal{X}$ using (4.12);
$p \leftarrow 1$;
while $p < L$ **do**
  Mask the values of the p least important features within $\mathcal{X}$;
  $\alpha_{simplified} \leftarrow$Re-evaluate the average network performance using the partially masked dataset ;
  if $|\alpha_{original} - \alpha_{simplified}| \geq \Delta$ **then**
    Eliminate the $p-1$ least important features;
    return *Subset of input state features*;
  end
  else
    $p \leftarrow p + 1$;
  end
end
Eliminate the $p-1$ least important features;
return *Subset of input state features*

the average network performance metric, ensuring that the algorithm outputs only a subset of importance-based features.

One major obstacle associated with the Deep-SHAP based SHAP_values estimation is the considerable computational complexity, which increases exponentially with the number of state features and linearly with the number of background data

samples [Lundberg and Lee, 2017]. Our initial experimentation shows that utilizing the whole training dataset $\mathcal{X}_{\text{BG}}$ significantly slows down the SHAP computations. To strike a balance between achieving high accuracy in SHAP_values estimates and keeping computation time reasonable, we utilize the transformation in the existing SHAP implementation by K-means to summarize $\mathcal{X}_{BG}$ [Lundberg et al., 2020]. Specifically, we use 1% K-means summary, i.e., 4,000 samples of $\mathcal{X}_{BG}$ and 10% K-means summary, i.e., 1,000 test samples, to serve as the background and hold-out dataset, respectively. This summarization is sufficient to accurately assign state feature importance to different input states without incurring the full computational burden of calculating SHAP_values over all data samples in the background dataset.

Note that the fast-changing nature of the state features does not pose a problem for the proposed XAI method. The post-training inference process can be performed in an environment twin (a simulation of the real environment), and the feature selection process is fast enough so that the environment does not change in the meantime. Additionally, our XAI-based feature selection process does not interact with the environment as it only utilizes the already-trained DRL agents and stored environment observations.

4.5 Performance Evaluation

This section evaluates the performance of the proposed MADRL resource allocation scheme in a single-cell V2X communications system. Numerical results are presented to validate the effectiveness of applying the XAI-based model simplification strategy.

The following algorithms are simulated for performance comparison to verify the efficiency of our proposed XAI-based simplified multi-agent DRL (Simplified-MADRL) scheme.

1. **Original multi-agent DRL (Original-MADRL)**: This algorithm trains the multiple V2V agents using the proposed centralized training with a decentralized execution framework described in Section 4.3.2. This algorithm considers the original full state features of the multiple agents during the training

and execution process, hence named Original-MADRL.

2. **Centralized Single-agent DRL (SADRL)**: This algorithm trains a single meta-agent at the BS, which outputs joint actions for all V2V agents. Since a single DQN jointly handles spectrum sub-band selection and transmit power control, the complexity of the output layer size and state action pairs to be visited for convergence is proportional to the total number of V2V pairs in the network. Apart from the complexity, this algorithm is executed in a centralized way with accurate global CSI availability requirements.

3. **Full power**: This algorithm selects the channel with the lowest interference for V2V transmission, setting the transmit power to the maximum level $P_{\max}$ for all V2V links. The computational complexity is dominated by the enumeration of all N available spectrum sub-bands, where a fast-sorting algorithm is used centrally at the BS to efficiently compare interference levels across the sub-bands and identify the one with the least interference [Li et al., 2022].

4. **Random allocation**: This algorithm chooses a random transmit power and spectrum sub-band action at the beginning of each time step.

For the learning algorithms, we consider an episodic setting with each episode lasting for $\mathcal{T}_{steps}$ =100 time steps. At the beginning of each episode, the environment state is initialized randomly, determined by the initial transmission powers of vehicular links and environment states. The training phase lasts $\mathcal{E}_{train} = 4000$ episodes, whereas $\mathcal{E}_{test} = 1000$ episodes are dedicated to the testing phase, where the well-trained DNNs are utilized to evaluate the performance of the different learning algorithms.

All simulations are performed on a 10-Core Intel(R) Xenon(R) Silver 4114, 2.2GHz system equipped with an Nvidia Quadro P2000 graphics processing unit (GPU). The V2X simulation environment and algorithms are developed in Python, and the DNNs are built and trained using Tensorflow API. The open-source imple-

mentation of SHAP [Lundberg et al., 2020] is used to compute explanations for the models.

4.5.1 Simulation Setup

We follow the simulation setup for V2X communications in annex A 3GPP TR 36.885, which describes in detail the vehicular channel models, vehicle mobility, and vehicular data traffic for an urban scenario [3GPP, 2023]. The urban scenario considers an area of size 750x1299 meters with 9 buildings and roads between buildings, with 2 lanes in each direction for each road. Vehicles are moving in this area at equal constant speed. N V2N links are initiated by N vehicles and the K V2V links are formed between each vehicle with its surrounding neighbors. The V2N links follow the path-loss model $PL(d) = 128.1 + 37.6\log_{10}(d)$, where d is the distance in [km] of the vehicle from the BS [Kyösti et al., 2007]. For the V2V links, we use the WINNER path-loss model $PL(d) = A\log_{10}(d) + B + C\log_{10}\left(\frac{f_c}{5.0}\right)$, where d is the inter-vehicular distance in [m], f_c is the system frequency in [GHz], the parameter A describes the impact of path-loss exponent, the parameter B is the intercept, and parameter C controls the frequency dependency. We set $\omega_1 = \omega_2 = 1$ in (4.7) to consider a fair and equal contribution of V2N and V2V link's sum rates. The parameter values for A, B, C are calculated according to the guidelines (Table 4-4) provided for $B1$ urban micro-cell scenario [Kyösti et al., 2007]. To achieve the demand of the URLLC, we set the packet transmission latency Δ_T to 1 ms, and the bandwidth occupied by each V2N link B_w to 180 KHz. Other important simulation parameters are provided in Table 4.1.

The hyperparameter values for the DQNs are tabulated in Table 4.2. The tabulated parameters are fine-tuned to reach the desired results and chosen as the final settings because they consistently deliver the best performance across multiple experimental trials. The initial learning rate $\delta^{(0)} = 0.01$ is annealed as $\delta^{(t+1)} = (1 - \beta_l)\delta^{(t)}$, with $\beta_l \in (0, 1)$ as the decay rate. The discount factor $\zeta = 0.99$ ensures higher cumulative rewards, benefiting V2V/V2N data transmission. The ϵ-greedy algorithm starts with $\epsilon^{(0)} = 0.1$ and adapts as $\epsilon^{(t+1)} = \max 0, (1 - \beta_\epsilon)\epsilon^{(t)}$,

Table 4.1: Simulation parameters

Parameter	**Value**
Carrier frequency	2 GHz
Maximum transmit power $P_{\max}$	23 dBm
Noise power σ^2	-114 dBm
Vehicle receiver noise figure	9 dB
Vehicle placement model	spatial Poisson
Carrier frequency f_c	2 GHz
Shadowing standard deviation σ_{sh}	3 dB
BS antenna height	25 m
BS antenna gain	8 dBi
BS receiver noise figure	5 dB
Distance between RSU and highway	35 m
Vehicle antenna height	1.5 m
Vehicle antenna gain	3 dBi
Payload size L	100 bits
Batch sampling period $\mathcal{T}_b$	100
Weight copy period $\mathcal{T}_c$	400

where $\beta_\epsilon = 0.0001$. Batch size $D = 100$ balances update noise and computational efficiency. The DNNs used to implement the DQN consist of one input layer, three fully connected hidden layers, and one output layer. The input layer processes a state vector of length $|\mathbf{x}| = (2+K) \times N$, while the hidden layers have $H_1 = 5|\mathbf{x}| + 8$, $H_2 = 3|\mathbf{x}|$, and $H_3 = 2|\mathbf{x}|$ neurons. This architectural design ingenuity reduces model size based on input feature reduction, balancing computational efficiency and accuracy. The rectifier linear unit (ReLU) and hyperbolic tangent (Tanh) functions are used alternatively as the activation function since this scheme leads to faster convergence, as per our observation.

4.5.2 Analysis of XAI-based Variable State Feature Selection

The main objective of our work is to simplify the proposed DRL-based framework through importance-based feature selection without losing performance. We first investigate the efficiency of the proposed importance-based variable feature selection methodology. Notice that the importance-based feature selection is a post-hoc process whereby the agents are first trained and tested using the Original-MADRL

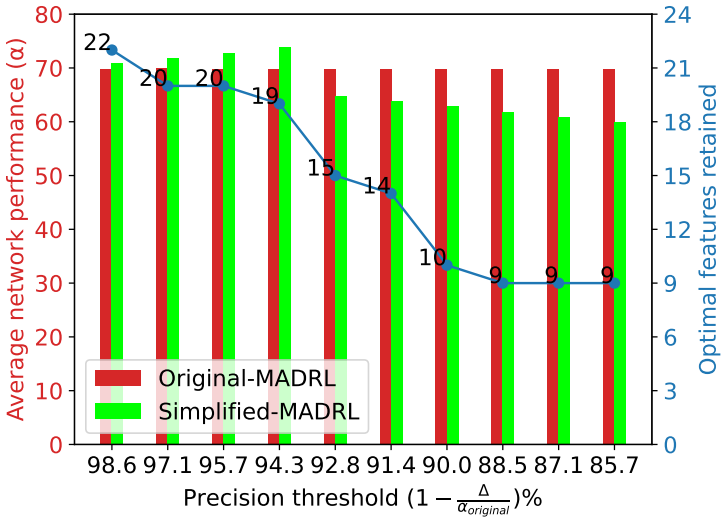
Table 4.2: Hyperparameters for Deep-Q Networks

Layers	Input	layer 1	layer 2	layer 3	Output
Neuron number	$(2+K)\times N$	$5\|\|\mathbf{x}\|\|+8$	$3\|\|\mathbf{x}\|\|$	$2\|\|\mathbf{x}\|\|$	$L_Q \times N$
Activation function	Linear	Tanh	ReLU	Tanh	ReLU
Mini-batch size D	100				
Optimizer	RMSProp				
Discount factor ζ	0.99				
Initial learning rate $\delta^{(0)}$	0.01				
Learning decay rate β_l	0.0001				
Initial exploration rate $\epsilon^{(0)}$	0.1				
ϵ-decay rate β_ϵ	0.0001				

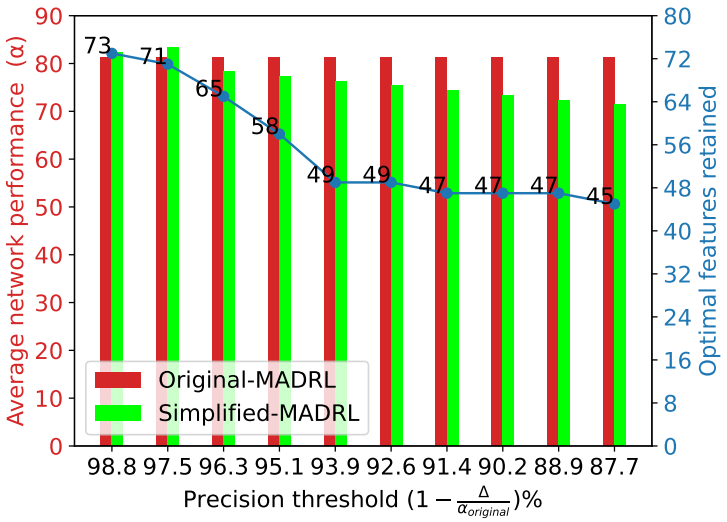
framework. Then, the summarized datasets $\mathcal{X}_{BG}$, $\mathcal{X}$ obtained from the training and execution process are utilized to obtain global state feature importance rankings using the systematic XAI-based methodology described in Section 4.4.1.

Fig. 4.4a shows the average network performance α and importance-based retained state features for different Δ values. Considering different QoS requirements for the V2V agents, we set different error probability constraints for each vehicle, $\varepsilon_{\max,k} = [10^{-4}, 10^{-3}, 10^{-4}, 10^{-3}]$. It can be observed that an increase in Δ allows for retaining fewer optimal features. As seen, for $\Delta \leq 2$, the Simplified-MADRL network does not lose any performance compared to the Original-MADRL network. In fact, a slight performance improvement is observed due to the elimination of less important features that are most likely non-informative and add complexity to the network without providing performance benefits. For $\Delta > 2$, there is a sharp decrease in the number of state elements, where approximately 38% of the original state features are eliminated. This sharp decline is reflected by the decrease in α, showing that some of the influential state input features have been removed.

Fig. 4.4b shows the average network performance and retained state features for different Δ values in a network with $N = 8$ V2N links and $K = 8$ V2V pairs. Here, we set error probability constraints as $\varepsilon_{\max,k} = [10^{-3}, 10^{-4}, 10^{-2}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-3}, 10^{-4}]$. Similar to the case for four V2V links, it can be observed that an increase in Δ re-



(a)



(b)

Figure 4.4: Average network performance and retained state features versus the precision threshold, with (a) $K = N = 4$ where the Original-MADRL network has 24 state features per agent, and (b) $K = N = 8$ where the Original-MADRL network has 80 state features per agent.

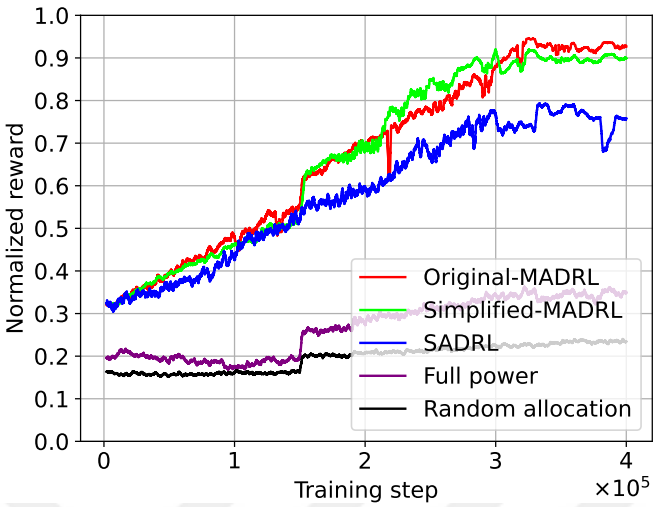
sults in fewer optimal state feature selections. For $\Delta = 2$, the proposed variable feature selection methodology selects 58 optimal state features out of 80 for the Original-MADRL network while maintaining the average network performance gap

less than ∼5% between the Original-MADRL scheme and the Simplified-MADRL scheme, respectively. Based on the results from Fig. 4.4a, and 4.4b, there exists a trade-off between model input simplification (e.g., via feature selection) and network performance. To strike the right balance between feature selection and optimal performance in terms of the average network performance metric, a moderate value of $\Delta = 2$ is selected to achieve performance close to the Original-MADRL network while reducing the number of optimal state features by ∼21% and ∼28% for a network with K=4 and K=8 V2V pairs, respectively. Therefore, for the subsequent simulation results, we fix the value of the precision threshold to $\Delta = 2$.

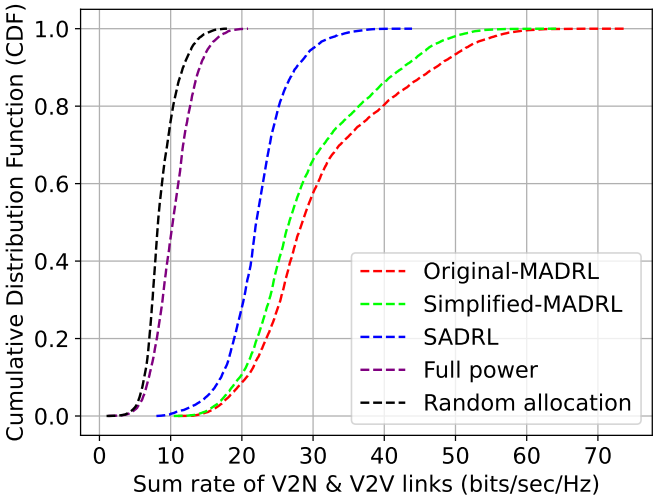
4.5.3 Performance Comparison of Algorithms

Convergence Analysis and Sum rate performance

Fig. 4.5a illustrates the training convergence behavior of the different algorithms in terms of the reward performance in a network with 4 V2V links and V2N links, with $\boldsymbol{\varepsilon}_{\max} = [\varepsilon_{\max,1}, \ldots, \varepsilon_{\max,K}] = [10^{-4}, 10^{-3}, 10^{-4}, 10^{-3}]$. It can be observed that the Original-MADRL scheme and the Simplified-MARL schemes converge around 3.2×10^5 and 2.8×10^5 time steps, respectively. In contrast, the centralized SADRL scheme demonstrates slow and suboptimal convergence, primarily due to the high complexity of its action space, where a single DQN determines power and sub-band allocation actions for all V2V links, rendering the approach impractical due to the significant transmission overhead involved. The Full power algorithm exhibits degraded performance as it always transmits at maximum power. Such resource allocation can lead to a lower sum rate and violation of reliability constraint if multiple links use the same sub-band. Moreover, the Full power algorithm does not consider adapting the transmit power based on the link quality. The random allocation exhibits the worst performance because it randomly selects the sub-band and transmits power at each time step, irrespective of the link quality or available resources. The proposed XAI based Simplified-MADRL scheme utilizing only 19 state features can remarkably converge to ∼97% of the training performance of the Original-MADRL using 24 state feature inputs. This means the proposed variable



(a) Training.



(b) Testing- Empirical CDF.

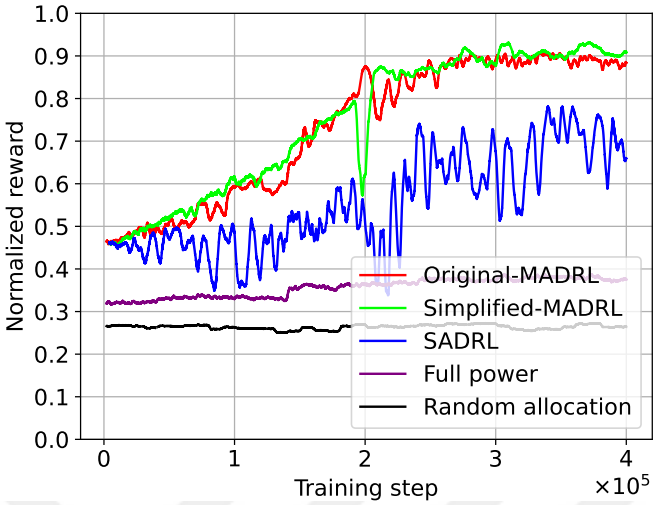
Figure 4.5: Training and testing performance of network with $K = N = 4$, where (a) shows the learning process comparison for the different algorithms, (b) shows the empirical CDF of network sum-rate performance. Each value is the moving average of the previous 300 time slots.

selection strategy can effectively filter the most relevant features contributing to the agents' actions without significant performance loss.

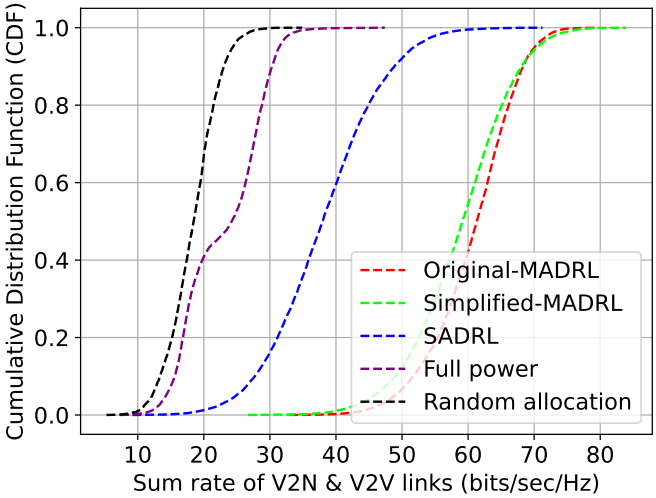
Fig. 4.5b shows the cumulative distribution function (CDF) of the achievable sum rate for the V2V and V2N links, comparing the performance of different algorithms during the deployment (inference) phase in a network with $K = N = 4$ V2V and V2N links. Fig. 4.5b further demonstrates the effectiveness of the proposed Simplified-MARL algorithm by empirically validating its performance against the different benchmark algorithms in the inference stage. The Simplified-MARL scheme outperforms the SADRL scheme and achieves on average 41% higher achievable sum-rates for all the V2V and V2N links compared to the SADRL scheme. Further, the average difference between the Simplified-MARL and the Original-MADRL schemes in terms of the sum-rate performance is less than $\sim$7%. This means that by exploiting the well-trained DNNs, the proposed algorithm can effectively learn the patterns of the environment based on the optimal state features without requiring the full set of state feature inputs.

Fig. 4.6a illustrates the training convergence behavior of different algorithms in a network with $K = N =$8 V2V and V2N links, with $\boldsymbol{\varepsilon}_{\max} = [\varepsilon_{\max,1}, \ldots, \varepsilon_{\max,K}] = [10^{-3}, 10^{-4}, 10^{-2}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-3}, 10^{-4}]$. It is observed that the rewards are low and unstable at the beginning of the learning as the training agents focus more on exploration than exploitation. Despite fluctuations due to mobility-induced channel fading in vehicular environments, the reward function increases and eventually converges with the increasing number of training iterations. The Simplified-MADRL and the Original-MADRL schemes have comparable training performance, and both schemes converge in about 2.8×10^5 training steps. The SADRL scheme fails to achieve comparable performance to the proposed method and demonstrates reduced training efficiency under higher vehicular densities. This inadequacy is attributed to the rapid growth of the DQN network topology and action space as the vehicular network scales, making it challenging to thoroughly explore the entire action space to determine the optimal action selection.

Similar to Fig. 4.5b, Fig. 4.6b shows the CDF of the achievable sum-rate for the V2V and V2N links; however, in a network with $K = N = 8$ V2V and V2N links. The proposed approach outperforms the SADRL scheme, achieving, on average,



(a)



(b)

Figure 4.6: Training and testing performance of the network with $K = N$ =8, where (a) shows the learning process comparison for the different algorithms, (b) shows the empirical CDF of network sum-rate performance. Each value is the moving average of the previous 300 time slots.

a 66% higher sum-rate in the testing stage. Interestingly, the Simplified-MADRL achieves higher sum-rates for the majority of the V2N and V2V links than the

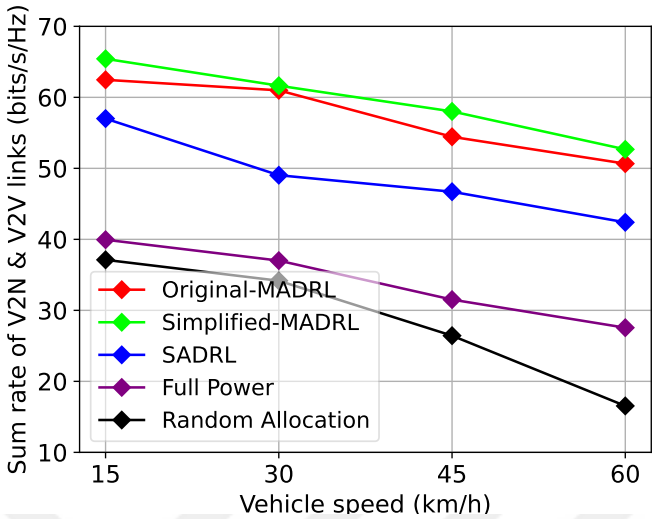


Figure 4.7: Sum rate performance of V2N and normal V2V links with varying vehicle speeds.

Original-MADRL scheme with ∼28% fewer state features. This improvement is mainly attributed to eliminating less important and non-informative features that only complicate the learning process.

Note that compared to the recursive feature selection strategies requiring re-training for different subsets of features, the desired results for the Simplified-MARL algorithm are achieved after a single retraining of the models on the subset of important features predicted by Algorithm 4. As seen from Fig 4.5b, 4.6b, the simplified agents have not lost significant performance when compared to the original agents' performance. In fact, a slight performance improvement in learning efficiency for larger network size is observed in Fig 4.6 due to eliminating less important features. This demonstrates that the proposed feature selection strategy effectively identifies the most relevant features for the multiple agent's actions with minimal impact on performance.

Impact of Mobility

Fig. 4.7 shows the impact of vehicle speed on the sum data rate of V2N and V2V links, when the number of V2V and V2N links are set to $K = N = 8$. From the figure, a general decline in sum rate performance is observed across all approaches as vehicle speed increases. This drop in performance is primarily due to reduced received power in V2X communication links, resulting from increased inter-vehicle distances at higher speeds, which lead to sparser traffic conditions. Additionally, higher speeds create observation uncertainties (e.g., channel state information and received interference), making it difficult to find an optimal policy—especially the random search approach, which shows the lowest performance. Our proposed Simplified-MADRL approach performs close to the Original-MADRL solution considering all state features during the learning process. Furthermore, the proposed MADRL-based solutions outperform the centralized SADRL and the Full power algorithms in terms of sum-rate performance across various vehicle speeds, adapting more effectively to dynamic networks.

Reliability performance

In Fig. 4.8, we consider the impact of reliability constraint violation. Thus, the performance metric is the network availability [She et al., 2018], defined as the percentage of the total number of channel generations in the testing phase, for which the worst-case decoding error probability, i.e., $\max_{k \in K} \varepsilon_k^{(t)}[n]$ is no greater than a threshold $\varepsilon_{\max}$. Here, we keep the same maximum threshold error probability for all V2V links, i.e, $\varepsilon_{\max,k} = \varepsilon_{\max}\ \forall k \in K$. Across all schemes, the network availability percentage improves with increasing $\varepsilon_{\max}$ as the tolerable decoding error probability requirement becomes less stringent. Original-MADRL consistently achieves the highest reliability, reaching ∼92% at $\varepsilon_{\max} = 10^{-5}$ for $K = N = 4$ and nearly 98.3% at $\varepsilon_{\max} = 10^{-4}$. The proposed Simplified-MADRL follows closely, with only a slight gap below Original-MADRL, achieving around 98% at $\varepsilon_{\max} = 10^{-4}$ for $K = N = 4$. Network availability decreases as the number of V2V and V2N links increases due to fixed resource sharing and higher interference, worsening reliability violations.

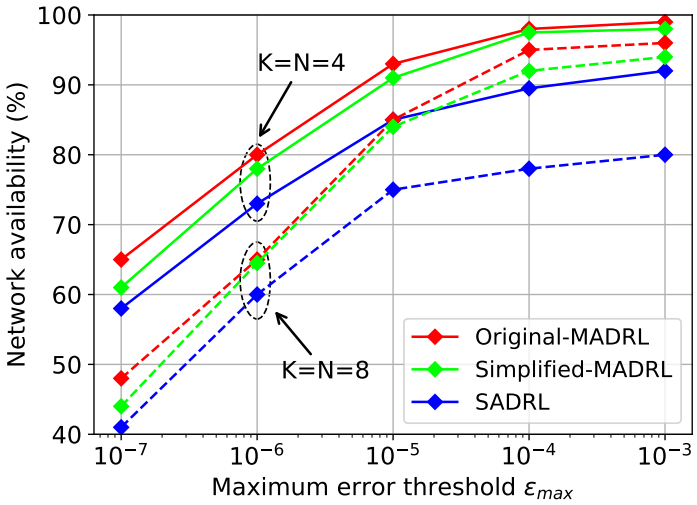


Figure 4.8: Network availability for different maximum error probability thresholds.

Simplified-MADRL maintains reasonable reliability for $\varepsilon_{\max} > 10^{-5}$, performing close to Original-MADRL. In contrast, centralized SADRL falls behind, achieving only $\sim 80\%$ for $K = N = 8$ at $\varepsilon_{\max} = 10^{-3}$. The gap is largest at lower error levels, highlighting the resilience of Original-MADRL and Simplified-MADRL to increased V2V link density.

Scalability

Fig. 4.9 demonstrates the impact of varying number of vehicles on the sum rate of the V2N links. Here, we fix the number of sub-bands for V2N links to $N = 4$ and increase the number of V2V links to assess the impact of dynamic vehicle numbers on the sum-rate performance. From the figure, it is observed that when the number of vehicles increases, more V2V links share the fixed number of sub-bands, which causes stronger interference to the V2N links, resulting in a rate reduction for all schemes. The proposed Simplified-MADRL approach achieves better performance than the other baseline schemes, demonstrating its robustness across different numbers of vehicles. The centralized SADRL approach shows degraded performance mainly caused by the increased size of the joint learning action space and increased DQN

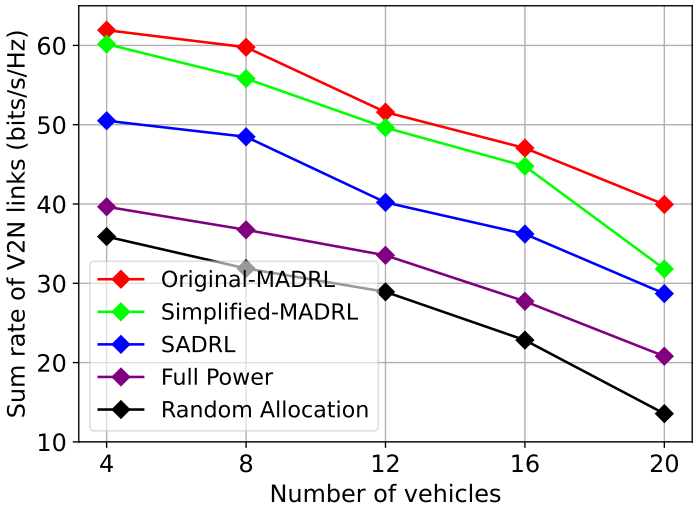


Figure 4.9: Sum rate of V2N links with varying number of vehicles.

output layer complexity, whereas the poor performance of the Full power algorithm can be explained by the strong interference due to maximum power transmission, negatively impacting the sum rate of the V2N links.

4.5.4 Computational Complexity Reduction Analysis

The computational complexity of the proposed learning algorithms depends on the structure of implemented DNNs (number of hidden layers and the number of corresponding neurons). The complexity of learning algorithms is primarily dominated by the update process of the network weight parameters. Each V2V agent requires a periodic update of its local DQN parameters during training. The input layer of each DQN always matches the number of input features, and the number of neurons in the hidden layers depends on these input features. Therefore, reducing the number of input features decreases the computational load for propagating information through the network and also reduces the number of parameters that need to be periodically broadcasted to update the local agents' DQNs.

The total number of parameters to broadcast per agent is $|\theta| = \left(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1}\right)$. The update process of the loss function is linear to the mini-batch size D since

Table 4.3: Complexity comparison of algorithms.

Scheme	**Average Training Time** (ms)		**Number of broadcast parameters**		**Number of state features**	
	K=4	K=8	K=4	K=8	K=4	K=8
Original-MADRL	6.23	13.4	34,544	347,920	24	80
Simplified-MADRL	4.56	11.9	22,424	185,912	19	58

the optimizer scans all mini-batch samples. As a result, an update of all parameters incurs a computational complexity of $\mathcal{O}\left(D(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1})\right)$. Considering the proposed Simplified-MADRL approach, the overall computational complexity is $\mathcal{O}\left(KD(\|\mathbf{x}\| + \sum_{l=1}^{3} H_l H_{l+1})\right)$ for updating the network parameters of all K agents. Table (4.3) shows the complexity comparison in terms of the average training time, the number of parameters to broadcast, and the number of state features used according to the variable feature selection methodology. Interestingly, the simplified-MADRL reduces average training time computed across training steps by ∼26% and ∼11% for K=4 and K=8 V2V agents, respectively. In terms of the the broadcast parameters per agent, the proposed approach achieves a reduction of ∼35% and ∼46% by utilizing 19 and 58 most important state features, compared to the 24 and 80 state features used by the Original-MARL agent. In our evaluation environment, the agents spend an average of 8.3×10^{-4} seconds for each action selection. There is considerable potential for optimizing execution efficiency since the computational capabilities of processing units in real-world V2X networks exceed those of the computers utilized in our simulations.

4.5.5 Discussion on Implementation and Future Extensions:

Practical implications: The proposed MADRL-based solution for spectrum and power resource allocation requires an offline training phase and a deployment phase. Offline training is employed to mitigate the computational overhead of learning the mapping between input states and output actions. Subsequently, during online network deployment, this mapping is executed using the pre-trained DNNs, bypassing

the need for intensive computations. Offline training can be performed using a digital twin of the V2X network, incorporating network topology, channel models, and QoS requirements. The proposed algorithm can utilize this twin for offline initialization and network training at the BS to explore the simulated environment, mitigating real-world risk. Further, in dynamic vehicular networks, it is challenging to provide theoretical convergence guarantees for the proposed MADRL-based solution and other advanced multi-agent adaptations, despite their good empirical performances [Hua et al., 2024]. By giving all agents a common global reward, the proposed MADRL-based solution mitigates the instability of the multi-agent environment but makes each agent fail to achieve the higher individual reward. The proposed XAI-based framework could be easily adapted to work with other multi-agent frameworks and cooperative value decomposition-based methods such as QMIX [Rashid et al., 2020] and its adaptations to improve convergence stability and cooperation among the agents.

Dynamic network topology: It is worth pointing out that the well-trained DRL models can be easily outdated due to the high mobility of vehicles. In our proposed centralized training and decentralized implementation approach, the weight parameters of the DNNs are centrally trained at the BS. The BS periodically broadcasts the latest trained weight parameters to the local agents. Retraining can be triggered if the network performance falls below a threshold. Further, considering nearby V2V pairs often experience similar channel quality and environment observations, the RSU can intelligently cluster vehicles into a specific zone and use the zone information to train newly activated V2V pairs. For newly activated V2V pairs, they request the BS for connectivity. The RSU has complete information on the vehicles' location and network topology. For an incoming vehicle, the RSU establishes the corresponding DNNs and transfers the weights of the already-trained model to the newly established DNNs, which are downloaded upon request from the incoming vehicle. To accommodate the changing number of V2V agents, state encoding adjusts for changes in the number of V2V agents under RSU coverage. If the number of vehicles exceeds the maximum number of trained agents, information from the extra

vehicles is not included in the state. Moreover, if one or more vehicles leave the coverage area, the RSU can introduce virtual agents with their direct and interfering channel gains set to zero to avoid impacting other agents' decision-making. The purpose of having these virtual agents as placeholders is to provide inconsequential inputs to fill the input elements of fixed length, i.e., the agent's state is padded with zeros. Further, the proposed XAI-based feature selection strategy somehow mitigates this issue since all trained agents use a fixed size of importance-based state features during training and testing.

Generalization to different environments: The proposed method is designed to adapt to the specific characteristics of a given vehicular environment. However, when the environment changes, the DNNs need to be retrained, and the feature selection needs to be reevaluated. In such cases, the multiple agents can be trained from scratch or initialized with the existing DNN weights, then refined using data from the new environment. To enhance the adaptability of the proposed MADRL-based solution to new environment scenarios and tasks requiring computation-intensive training, model-agnostic meta-learning [Yuan et al., 2021, Finn et al., 2017] can be incorporated into the proposed framework.

4.6 Conclusions

In this paper, we have developed a model-agnostic XAI-based methodology to explain the inference process of multiple trained DRL agents and have used it to simplify the DRL agents' input state size through importance-based feature selection. We have capitalized on SHAP to generate feature importance explanations and reduce the complexity of multiple trained models by generating a new model with a reduced size that requires lower training time and reduced network parameters to broadcast while maintaining reasonable performance. We have applied the proposed methodology to a practical vehicular communication network where multiple agents are first trained using a novel centralized learning and decentralized execution algorithm. Subsequently, the XAI-based framework is applied to the trained models to select important features using an automated feature selection strategy. Our re-

sults have shown that the XAI-assisted methodology remarkably reduces the optimal state features by ∼21% and ∼28%, average training time by ∼26% and ∼11%, and trainable weight parameters to broadcast by ∼35% and ∼46% in a network setting with four and eight vehicular pairs, respectively.

Chapter 5

EVENT-TRIGGERED REINFORCEMENT LEARNING BASED JOINT RESOURCE ALLOCATION FOR ULTRA-RELIABLE LOW-LATENCY V2X COMMUNICATIONS

5.1 Background and Overview

Vehicle-To-Everything (V2X) networks have recently attracted a lot of attention as a crucial enabler for intelligent transportation systems (ITS) by enhancing traffic safety, improving vehicle coordination, and enabling novel transportation features such as platooning, advanced real-time navigation, and lane change alerts. Compared to dedicated short-range communications (DSRC) and the long-term evolution (LTE) technology, the 5G New Radio (5G-NR) V2X offers several advantages, including controllable latency, larger coverage area, robust scalability, and high data rates, even in high-mobility scenarios [Bagheri et al., 2021]. While existing and predecessor technologies for V2X applications are mainly designed to cater to data-intensive applications for heterogeneous traffic and infotainment services (video streaming, location tracking, map updates), the ultra-reliable and low-latency communications (URLLC) aspect is not adequately studied to support safety-critical services such as cooperative adaptive cruise control and intersection collision warnings. Advanced V2X use cases, e.g., vehicle platoon, advanced driving, extended sensors, real-time navigation, and remote driving often demand a transmission latency within a few milliseconds and reliability on the order of 99.999% [Yang et al., 2020b] [Durisi et al., 2016].

To meet the stringent quality-of-service (QoS) requirements in the URLLC network design, the control messages are required to be transmitted with short packet

size (e.g., 50 $\sim$ 300 bytes) [Dong et al., 2022]. Most existing works on V2X communications are based on Shannon's capacity theorem [Ding et al., 2022, Sun et al., 2016, Wu et al., 2021] implicitly assuming infinite block length availability. However, for the practical realization of URLLC-based V2X applications, the information size is limited, which necessitates incorporating the finite block length (FBL) information theory to reduce the transmission delay. Additionally, for small packet sizes, the decoding error probability is no longer negligible, and the trade-off between the transmission rate, decoding error probability, and transmission latency (in terms of block length/channel uses) cannot be captured by Shannon's capacity formula. As an important enabler of URLLC, recent works have adopted the FBL structure for resource allocation in V2X networks with the objective of network-wide power minimization [Abdel-Aziz et al., 2020], the transmission information maximization [Yang et al., 2020a], and the worst-case transmission latency minimization [Dong et al., 2022], [Fu et al., 2021]. However, these works mainly focus on achieving ultra-low latency, whereas the reliability aspect in terms of minimizing the decoding-error probability is not considered. Additionally, these works [Ding et al., 2022, Sun et al., 2016, Wu et al., 2021, Yang et al., 2020a, Dong et al., 2022, Abdel-Aziz et al., 2020, Fu et al., 2021] overlook optimizing the block length, which significantly impacts the delay performance of latency-sensitive communications.

The minimization of decoding-error probability has been investigated previously in wireless resource allocation schemes to characterize the reliability performance in the design of energy harvesting networks [Hu et al., 2019], unmanned aerial vehicle (UAV) assisted communication networks [Pan et al., 2019], and industrial automation systems [Ren et al., 2020a], [Nasir, 2020]. In [Hu et al., 2019], the simultaneous wireless information and power transfer (SWIPT) system parameters and block length are concurrently optimized to increase the system's reliability. The location and block length of a UAV are jointly optimized while guaranteeing bounds on the target decoding-error probability in [Pan et al., 2019]. However, these works do not consider optimizing the power allocation, which has a significant impact on the reliability of the system. In [Ren et al., 2020a], the problem of improving the target

device reliability while dealing with the transmit power and block length limitations is addressed in an industrial automation scenario. It is demonstrated that the decoding-error probability set for one device determines the achievable error probability performance for the other device, leading to unfair reliability across various devices. [Nasir, 2020] focuses on achieving fair reliability among the devices by minimizing the maximum decoding-error probability for power allocation and transmission block length optimization in an industrial automation scenario. The aforementioned works [Hu et al., 2019, Pan et al., 2019, Ren et al., 2020a, Nasir, 2020] do not investigate the joint convexity of decoding-error probability with respect to the block length and transmit power, which can further improve the performance of systems requiring bounded latency and extreme reliability [Zhu et al., 2022]. Additionally, these works adopt iterative optimization theory-based algorithms inherently experiencing high runtime complexity, making them impractical for mission-critical V2X applications.

Considering that the time frame-based decision-making can be regarded as a Markov decision process (MDP), many studies have applied reinforcement learning methods for resource allocation in V2X networks. Reinforcement learning (RL) effectively addresses decision-making under uncertainty and solves hard-to-optimize objective functions efficiently [Luong et al., 2019]. Deep RL (DRL) combines RL with deep neural networks (DNN) to deal with the large state-action space, making it particularly appealing for real-time resource management. DRL has been widely adopted to address resource allocation problems in URLLC-aware V2X networks scenarios [Li et al., 2022, Zhang et al., 2020, Ye et al., 2019, Guo et al., 2023, Xiang et al., 2021]. [Li et al., 2022] proposes a federated multi-agent DRL scheme for channel selection and power allocation in a decentralized fashion by incorporating the reliability constraint in terms of the signal-to-interference-plus-noise ratio (SINR) outage probability on the vehicle-to-vehicle (V2V) communication links. [Zhang et al., 2020] considers a two-timescale federated DRL algorithm combined with spectral clustering for transmission mode selection in a V2X communication scenario, where the reliability and latency constraints in terms of tolerable outage

probability are incorporated into the constrained optimization problem. A multi-agent DRL scheme to allocate resources for both unicast and broadcast V2X applications in the absence of the global channel state information (CSI) is addressed in [Ye et al., 2019], where the URLLC constraints are incorporated into the reward function directly. [Guo et al., 2023] proposes a hybrid centralized-distributed DRL-based resource allocation scheme by considering both the total system capacity and reliability of the vehicular links in the design objective. [Xiang et al., 2021] formulates a multi-objective resource allocation problem to maximize the transmission success ratio and the mean opinion score of vehicular links for intra-platoon communications using a novel DRL-based framework, where the latency requirement is incorporated into the observation space of the DRL agent as the percentage of the remaining time over the transmission latency. For the mentioned literature on URLLC-enabled V2X systems [Li et al., 2022, Zhang et al., 2020, Ye et al., 2019, Guo et al., 2023, Xiang et al., 2021], only V2V communication is utilized for disseminating crucial safety messages among vehicles, whereas the vehicle-to-infrastructure (V2I) links are dedicated to data-intensive services and non-safety infotainment applications using infinite block length transmissions. Nevertheless, the reliability of the V2V links diminishes when considering blockage effects, which hampers the performance of V2V communications. Additionally, advanced URLLC-based V2X use cases (e.g., teleoperated driving and vehicle platooning) often rely on V2I links for remote control [Garcia et al., 2021b], [Fang et al., 2022]. Moreover, the resource allocation strategies [Guo et al., 2023, Xiang et al., 2021, Li et al., 2022, Zhang et al., 2020, Ye et al., 2019] may not be suitable for realistic URLLC-6G resource management problems in actual wireless systems because the learning process is conducted in a time-triggered mechanism, i.e., the resource allocation decisions are updated periodically in each time frame, requiring computationally intensive training procedures.

Against the time-triggered mechanism, event-triggered learning has been proposed with the goal of activating computations or communications only when the system state deviates from the expected accuracy or when the triggering condition is

violated [Solowjow and Trimpe, 2020]. Recently, event-triggered learning and DRL have been combined for path planning in an autonomous driving scenario [Lu et al., 2023]. The proposed methodology involves incorporating the triggering condition as communication loss into the reward function, learns the driving control policy for autonomous path planning, and is able to improve the system performance compared to a standard DRL-based learning approach. None of these studies however incorporate event-triggered learning into a DRL approach for minimizing the computational expenses and optimizing the continuous power and integer block length allocation in a URLLC-aware V2X network.

5.1.1 Contributions and organization

This paper proposes novel adaptive solution strategies to jointly optimize the transmit power and block length allocation with the objective of minimizing the worst-case decoding-error probability among all the vehicles subject to the stringent reliability and latency requirement for a time-critical V2X URLLC scenario. A preliminary version of this work proposing a sub-optimal algorithm based on the block coordinate descent (BCD) method for the block length and transmit power optimization in a vehicular communication scenario is described in [Khan and Coleri, 2022]. Different from our previous work [Khan and Coleri, 2022], we first analyze the joint-convexity of the optimization problem with respect to the block length and transmit power variables, derive the Karush–Kuhn–Tucker (KKT) optimality conditions to solve the joint optimization problem, and analyze its applicability in a vehicular network. Additionally, we propose a DRL-based robust algorithm for the block length assignment and power allocation problem, which can effectively handle the excessive computational complexity associated with the joint optimization scheme. Further, we incorporate event-triggered learning into the proposed DRL-based framework enabling the assessment of whether to initiate the DRL process. This, in turn, leads to a decrease in the frequency of DRL process executions, all the while upholding acceptable levels of reliability performance.

The original contributions of this chapter are listed as follows:

- We propose a novel optimization theory-based solution strategy for the joint optimization of code block length and transmit power with the goal of minimizing the maximum decoding error probability of all vehicles while considering the transmit power, reliability, and latency constraints in a URLLC-enabled vehicular network, for the first time in the literature. We show that the decoding error probability function preserves joint convexity in block length and transmit power within reasonable limits on the block length. We then relax the integer block length constraint, derive optimality conditions by using the KKT analysis, solve the problem by using Lagrangian dual decomposition method and utilize a low-complexity greedy search methodology to convert the positive continuous block length values to an integer solution.

- We propose a centralized event-triggered DRL-based scheme to determine the optimal block length and transmit power allocation policies to achieve fair reliability while satisfying the URLLC constraints, for the first time in the literature. For the DRL scheme, a two-layered training framework is designed. In the first layer, a multi-DQN structure is used separately for code block length assignment. The second layer involves an actor-critic network and utilizes the deep deterministic policy-gradient (DDPG)-based algorithm to optimize the transmit power allocation. To accelerate learning efficiency and relieve the trainer of computation stress, we leverage event-triggered learning by introducing a trigger block based on the input state similarity.

- Via extensive simulations, we demonstrate that our proposed event-triggered DRL-based approach outperforms the proposed joint optimization method and the recent state-of-the-art DRL-based approach utilizing a single-DQN to optimize the joint resource allocation in terms of computational complexity and convergence time for different network configurations.

5.2 System Model and Assumptions

We consider a single-cell V2I URLLC system, where vehicles move on a multi-lane highway with the RSU at the center and a fixed distance from the highway, as shown in Fig. 5.1. The roadside unit (RSU) plays the role of the 5G base station (gNB) for centralized radio resource management [Garcia et al., 2021b] and sends unicast safety-critical information to the vehicles for coordination and safety alerts using short packets. We consider a practical scenario where this RSU is deployed in a rural area and deprived of a permanent grid source. We assume the RSU is equipped with large batteries that are periodically recharged using energy harvesting techniques (wind/solar) [Atallah et al., 2016]. We assume that the RSU and the vehicle user equipment (VUEs) are equipped with a single transmit-receive antenna. The set of VUEs under the coverage of the RSU is denoted by $\mathcal{K} = \{1, \ldots, K\}$. This system model represents different URLLC V2X scenarios, such as time-sensitive High-Definition (HD) map data dissemination for vehicle coordination and localization in autonomous driving, coordinating the trajectories of vehicles to make better maneuvers in remote/teleoperated driving, and cooperative collision avoidance for vehicle safety in practical traffic management scenarios [Alalewi et al., 2021].

In NR-V2X mode-1 operation, the RSU centrally allocates the resources for V2X communications. This mode requires the vehicles to report their CSI to the gNB when initiating the V2X communications. The RSU transmits the CSI reference signal (CSI-RS) through the Physical downlink Control Channel (PDCCH) for downlink CSI acquisition. Then, the VUE decodes the received CSI-RS to acquire channel measurements. Subsequently, the VUE reports the CSI to the RSU using the Physical Uplink Control Channel (PUCCH). The CSI transmission introduces additional overhead, which can be mitigated by incorporating sensing-assisted communications [Li et al., 2023]. In this approach, the RSU can directly acquire the downlink CSI based on the vehicles' echo signals, capturing the vehicles' motion parameters from the echo and analyzing the channel quality based on the power of

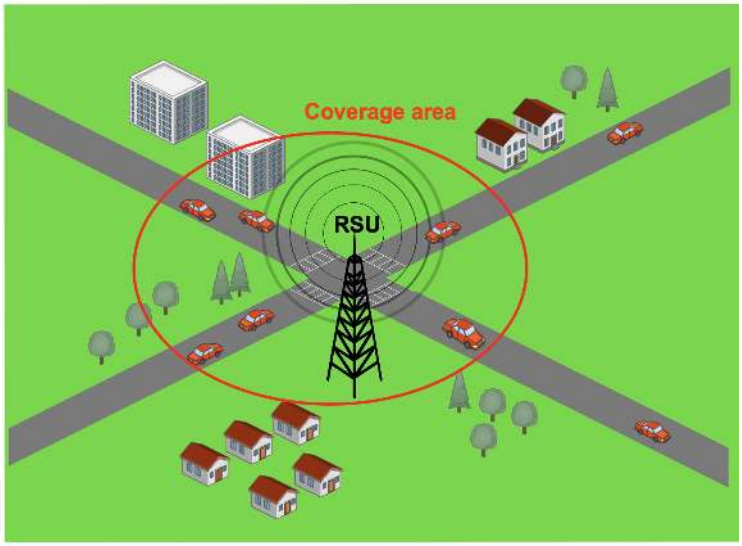


Figure 5.1: System model for the V2X communication network.

the echo signal.

We consider a slotted communication system with the time period for information transmission discretized into a series of time frames $t \in \{1, 2, \cdots, \mathcal{T}\}$. The URLLC-aware resource allocation algorithm sends K short data packets to the K VUEs in each time frame, and the transmission of these packets is handled in a time-division-multiple-access (TDMA) fashion such that each VUE is allocated with unique block lengths. To fulfill the requirements of latency-sensitive communications, scheduling is executed at the beginning of each time frame. The entire downlink information transmission is subject to strict restrictions on the latency, requiring completion within a single time frame spanning M_D symbols such that $\sum_{k \in \mathcal{K}} m_k^{(t)} \leq M_D$, where $m_k^{(t)}$ is the channel block length allocated to the k^{th} VUE. Here, M_D represents the maximum block length, indicating the maximum number of symbols transmitted within a time frame, and ensures timely delivery of all short packets to vehicles within the maximum transmission delay $T_{max} = \frac{M_D}{B}$, where B is the system bandwidth. This aggregate delay assumption ensures that the packets generated for all the VUEs during a time frame are scheduled in the next frame, and this assumption is considered in several other works [Pan et al., 2019, Ren et al., 2020a, Nasir, 2020].

Moreover, the continuous transmit power allocated to the k^{th} VUE in time frame t, denoted by $P_k^{(t)} \geq 0$, is subject to the constraint $\left(\sum_{k=1}^{K} \frac{P_k^{(t)} m_k^{(t)}}{M_D}\right) \leq P_{\max}$, where $\frac{m_k^{(t)}}{M_D}$ is the fractional block length allocated to the k^{th} VUE, and $P_{\max}$ is the RSU power budget.

The downlink channel gain from the RSU to the k^{th} VUE in time frame t is denoted by $h_k^{(t)} = \alpha_k^{(t)} \left| g_k^{(t)} \right|^2$, where $\alpha_k^{(t)}$ denotes the large-scale fading (path loss and log-normal shadowing) and $g_k^{(t)}$ denotes the complex Gaussian random variable with Rayleigh distributed envelope. We utilize a quasi-static flat fading channel model where the channel remains constant in the coherent time $\mathcal{T}_c$ but might change with correlation patterns across consecutive time frames. The channel coherence time is given by $\mathcal{T}_c = \sqrt{\frac{9}{16\pi f_D^2}}$, where $f_D = \frac{f_c \cdot v_s}{c}$ is the Doppler frequency, f_c is the carrier frequency, v_s is the vehicle velocity, and c is the speed of light [Li et al., 2024]. For moderate vehicular speeds, the transmission time interval, considered equal to the duration of one frame, is far less than the channel coherence time, i.e., $\mathcal{T}_c > t$ holds. Thus, the CSI can be regarded as constant throughout the information transmission period, i.e., the channel remains static over M_D symbols. We use a first-order Gauss-Markov process to describe the relationship between the small-scale Rayleigh fading in two consecutive time frames according to Jake's statistical model:

$$g_k^{(t)} = \rho g_k^{(t-1)} + w_k^{(t)}, \tag{5.1}$$

where $\rho\,(0 \leq \rho \leq 1)$ is the correlation coefficient between two consecutive small-scale fading realizations, ${w_k}^{(t)} \sim \mathcal{CN}\,(0, 1-\rho^2)$ is the channel discrepancy term, $g_k^{(0)} \sim \mathcal{CN}(0,1)$ and $\mathcal{CN}$ represents a circularly symmetric complex normal Gaussian random. The correlation coefficient is given by $\rho = J_0\,(2\pi f_D T)$, where $J_0(.)$ is the zeroth-order Bessel function, and T is the time interval over which the correlated channel variation occurs.

In the context of finite block length regime, the achievable information rate for the k^{th} VUE, which can be decoded with decoding error probability no greater than

$\varepsilon_k^{(t)}$ is given by the normal approximation [Polyanskiy et al., 2010]

$$R_k^{(t)} \approx \left(\log_2\left(1+\gamma_k^{(t)}\right) - \sqrt{\frac{V_k^{(t)}}{m_k^{(t)}}} \frac{f_Q^{-1}\left(\varepsilon_k^{(t)}\right)}{\ln 2} \right), \tag{5.2}$$

where $V_k^{(t)} \triangleq 1-(1+\gamma_k^{(t)})^{-2}$ is the channel dispersion, $\varepsilon_k^{(t)} \in (0,1)$ is the decoding error probability, $\gamma_k^{(t)} = \left(\frac{P_k^{(t)} h_k^{(t)}}{\sigma^2}\right)$ is the signal-to-noise-ratio (SNR) for the k^{th} VUE in time frame t, σ^2 is the noise power and $f_Q^{-1}(\cdot)$ is the inverse of the function $f_Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}}\, dt$. Eq. (5.2) is a non-asymptotic approximation for the achievable rate for finite block length codes. Practical coding schemes (e.g., extended BCH, LDPC, Polar codes) closely approach the normal approximation while maintaining a relatively constant gap in the block error rate to the finite block length fundamental limit [Shirvanimoghaddam et al., 2019]. Therefore, we consider Eq. (5.2) a tight and accurate approximation for the maximal coding rate under finite block length transmissions.

We consider a fixed payload size of L bits per vehicle. For a given block length $m_k^{(t)}$, the coding rate for the k^{th} VUE in time frame t becomes $R_k^{(t)} = \frac{L}{m_k^{(t)}}$ [bits/seconds/Hz]. By rearranging Eq. (5.2), the decoding error probability at the k^{th} VUE can be expressed as

$$\varepsilon_k^{(t)}\left(\gamma_k^{(t)}, m_k^{(t)}, L\right) \approx f_Q\left(\frac{\sqrt{m_k^{(t)}} \ln 2}{\sqrt{V_k^{(t)}}} \left(C_k^{(t)} - \frac{L}{m_k^{(t)}} \right) \right), \tag{5.3}$$

where $C_k^{(t)} = \log_2\left(1+\gamma_k^{(t)}\right)$ is the Shannon rate. For typical URLLC-enabled V2I communications for traffic efficiency/safety, the codewords are required to be short [Li et al., 2024], which can easily satisfy the URLLC requirement subject to the availability of sufficient SNR. When the instantaneous CSI of the downlink channel is poor, i.e., $\gamma_k \leq \gamma_{th} = 0$ dB, the reliability performance cannot be guaranteed even with maximally available block length. In such a scenario, we assume that the RSU does not allocate resources and leaves it to the next time frame.

5.3 Worst-case Decoding Error Probability Minimization Problem

In this section, we present the mathematical formulation of the *Min-Max* decoding-error probability problem denoted by $\mathcal{MMDEP}$. In this section, we omit the index (t) for notational simplicity.

The joint optimization of the transmit power and block length allocation to minimize the worst-case decoding error probability under stringent latency and reliability constraints is formulated as follows:

$\mathcal{MMDEP}$:

$$\min_{\boldsymbol{P},\boldsymbol{m}} \quad \max_{k\in K} \ \varepsilon_k \tag{5.4a}$$

$$\textit{subject to} \qquad \sum_{k=1}^{K} \frac{m_k P_k}{M_D} \leq P_{max} \tag{5.4b}$$

$$\sum_{k=1}^{K} m_k \leq M_D, \tag{5.4c}$$

$$\varepsilon_k \leq \varepsilon_{max,k},\ k \in \{1, ..., K\} \tag{5.4d}$$

$$\textit{variables} \qquad P_k \geq 0,\ m_k \in \mathbb{Z}^+. \tag{5.4e}$$

where $\boldsymbol{P}$ and $\boldsymbol{m}$ are the transmit power and block length allocation vectors of the network, i.e., $\boldsymbol{P} = [P_1, \ldots, P_K]$ and $\boldsymbol{m} = [m_1, \ldots, m_K]$, and $m_k \in \mathbb{Z}^+$ indicates the set of all positive integers. The variables of the optimization problem are P_k, the transmit power allocated to the k^{th} VUE; m_k, the block length in units of symbols or channel uses allocated to the k^{th} VUE, $\forall k \in \{1, \cdots, K\}$.

The objective of the optimization problem is to minimize the worst-case decoding error probability for the vehicular network as given by (5.4a). (5.4b) represents the total power budget constraint, ensuring that the sum of the product of fraction of the time allocated to each VUE and its corresponding power allocation is within the total power budget of the RSU. (5.4c) gives the latency constraint: The total block length assigned to all the VUEs should be less than or equal to the maximum number of symbols allowed to conduct the information transmission. To ensure the quality of transmission, (5.4d) represents the upper bound of tolerable packet error probability for each VUE, with $\varepsilon_{max,k}$ denoting the corresponding quality of service

requirement for the k^{th} vehicle. Finally, (5.4e) represents a non-negative transmit power constraint and the positive integer constraint for the block length assigned to VUE.

The optimization problem under consideration is a non-convex mixed integer nonlinear programming (MINLP) due to the non-convex objective function (5.4a), non-convex constraint (5.4b) and the integer constraint for the block length (5.4e). Therefore, directly achieving the global optimal solution is extremely difficult, i.e., finding a global optimum requires algorithms with exponential complexity. Next, we propose an optimization theory based solution and an event-triggered DRL-based framework to solve this problem.

5.4 Joint Optimization of Power and block length

In this section, we reformulate the non-convex optimization problem $\mathcal{MMDEP}$ using the variable transformation method and relaxation of the integer block length constraint. The reformulated optimal power and block length problem is denoted by $\mathcal{OPBP}$. We show that within the region of interest, $\mathcal{OPBP}$ is jointly convex with respect to block length and transmit power as decision variables. We then derive the optimality conditions using KKT analysis and propose a Joint Optimization Algorithm ($\mathcal{JOA}$) to solve this problem. The $\mathcal{JOA}$ efficiently solves $\mathcal{OPBP}$ using standard convex optimization tools. It further employs a low-complexity suboptimal algorithm to convert the block lengths into integer values.

To facilitate the derivation for solving the $\mathcal{OPBP}$, we first relax the integer constraint for block length in (5.4e), i.e., $m_k \in \mathbb{R}^+$. Second, we exploit the variable transformation method, which is used to illustrate the joint-convexity of $\mathcal{OPBP}$ in the decision variables. In particular, we define $a_k \triangleq \frac{1}{m_k}$ and $b_k \triangleq \sqrt{P_k}$. Additionally, to handle the non-differentiable min-max objective function, we apply the epigraph transformation and minimize an upper-bound η as an alternate objective for minimizing the maximum decoding-error probability. This transformation facilitates the solution design while keeping the original objective and constraints intact. Then, the $\mathcal{OPBP}$ can be mathematically reformulated as follows:

$\mathcal{OPBP}$:

$$\min_{\boldsymbol{a},\boldsymbol{b},\eta} \quad \eta \tag{5.5a}$$

$$\textit{subject to} \quad \sum_{k=1}^{K} \frac{b_k^2}{a_k} \leq P_{max} M_D \tag{5.5b}$$

$$\sum_{k=1}^{K} \frac{1}{a_k} \leq M_D, \tag{5.5c}$$

$$\varepsilon_k\left(a_k, b_k\right) \leq \eta,\ k \in \{1, ..., K\} \tag{5.5d}$$

$$\varepsilon_k\left(a_k, b_k\right) \leq \varepsilon_{max,k},\ k \in \{1, ..., K\} \tag{5.5e}$$

$$\textit{variables} \quad a_k \geq 0,\ b_k \geq 0,\ \eta \in [0,1],\ k \in \{1, ..., K\}. \tag{5.5f}$$

where $\boldsymbol{a} = [a_1, \ldots, a_K]$ and $\boldsymbol{b} = [b_1, \ldots, b_K]$, and η, a_k and b_k, $\forall k \in \{1, \ldots, K\}$ are the decision variables of $\mathcal{OPBP}$. To this end, we first provide the following Lemma as a first solution step for solving the $\mathcal{OPBP}$.

Lemma 1 *Under the assumption that $\gamma_k\left(b_k\right) \geq \gamma_{th} = 0$ dB and the Shannon rate exceeds the coding rate, i.e., $C_k\left(\gamma_k\left(b_k\right)\right) - a_k L > 0$, the $\mathcal{OPBP}$ is a convex optimization problem when m_k satisfies:*

$$m_k \leq \min\left\{5\ln\left(2\right)\gamma_k\left(b_k\right)L\,,\ \frac{3\ln\left(2\right)10\sqrt{10\gamma_k\left(b_k\right)}L}{4}\right\}, \tag{5.6}$$

Proof: Since $f_Q(x)$ is a decreasing function of its argument and $\varepsilon_k < \varepsilon_{\max,k} < f_Q(0) = 0.5$, a necessary condition for constraint (5.5e) to hold is $C_k\left(\gamma_k\left(b_k\right)\right) - a_k L > 0, \forall k \in \mathcal{K}$. Further, let us define $\beta\left(a_k, b_k\right) \triangleq \left(\frac{\ln 2}{\sqrt{a_k}\sqrt{V_k(b_k)}}\left(C_k\left(\gamma_k\left(b_k\right)\right) - a_k L\right)\right)$. Then, $\beta\left(a_k, b_k\right)$ is non-negative and the second-order derivative of $\varepsilon_k\left(a_k, b_k\right) = Q\left(\beta\left(a_k, b_k\right)\right)$ is given by

$$\frac{\partial^2 \varepsilon_k\left(a_k, b_k\right)}{\partial \beta^2} = \frac{\beta}{\sqrt{2\pi}} e^{-\frac{\beta^2}{2}} > 0. \tag{5.7}$$

Therefore, the decoding error probability decreases monotonically and is a convex function with respect to β. Furthermore, under the condition that $\gamma_k\left(b_k\right) \geq \gamma_{th} = 0$ dB, the first principal minor and the determinant of the Hessian matrix of $\varepsilon_k\left(\beta\left(a_k, b_k\right)\right)$ are non-negative when (5.6) is satisfied . Therefore, $\varepsilon_k\left(\beta\left(a_k, b_k\right)\right)$ is

jointly convex in the auxiliary variables a_k and b_k in the considered high reliability scenario. Consequently, the $\mathcal{OPBP}$ is a convex optimization problem considering the linearity of the objective (5.5a) and convexity of the constraints (5.5b)-(5.5e) in variables a_k, b_k and η. $\square$

As $\mathcal{OPBP}$ is a convex optimization problem, it can be solved by analyzing its KKT conditions to reach an optimal solution. The corresponding Lagrangian function for (5.5) can be written as

$$\begin{aligned} L\left(a_k, b_k, \eta, \lambda_{i,k}\right) = & \Bigg[\eta + \lambda_1 \left(\sum_{k=1}^{K} \frac{b_k^2}{a_k} - P_{max} M_D \right) + \\ & \lambda_2 \Big(\sum_{k=1}^{K} \frac{1}{a_k} - M_D \Big) + \sum_{k=1}^{K} \lambda_{3,k} \left(\varepsilon_k - \eta\right) + \\ & \sum_{k=1}^{K} \lambda_{4,k} \left(\varepsilon_k - \varepsilon_{max,k}\right) \Bigg]. \end{aligned} \tag{5.8}$$

Then, the KKT conditions of $\mathcal{OPBP}$ are as follows:

$$\begin{aligned} & \Bigg\{ (\lambda_{3,k} + \lambda_{4,k}) \left(\frac{m_k^2 \ln 2 \exp \frac{-\beta^2}{2}}{\sqrt{2\pi}} \right) \left(\frac{1}{2} m_k^{-\frac{1}{2}} V_k^{-\frac{1}{2}} C_k + \frac{1}{2} m_k^{-\frac{3}{2}} V_k^{-\frac{1}{2}} L \right) \\ & - \lambda_1 \left(\sum_{k=1}^{K} P_k m_k^2 \right) - \lambda_2 \left(\sum_{k=1}^{K} m_k^2 \right) \Bigg\} = 0, \;\; k \in \{1, ..., K\}, \end{aligned} \tag{5.9a}$$

$$\begin{aligned} & \Bigg\{ (\lambda_{3,k} + \lambda_{4,k}) \left(\frac{\sqrt{m_k}\sqrt{P_k}(H_k^3 P_k^2 + 2H_k^2 P_k - H_k C_k) + \frac{L\sqrt{P_k} H_k}{\sqrt{m_k}}}{(1 + H_k P_k)^3} \right) \\ & \times \left(\frac{-2 \ln 2 V_k^{-\frac{3}{2}} \exp \frac{-\beta^2}{2}}{\sqrt{2\pi}} \right) + \lambda_1 \left(2 \sum_{k=1}^{K} \sqrt{P_k} m_k \right) \Bigg\} = 0, \;\; k \in \{1, ..., K\}, \end{aligned} \tag{5.9b}$$

$$\left\{ 1 - \sum_{k=1}^{K} \lambda_{3,k} \right\} = 0, \;\; k \in \{1, ..., K\}, \tag{5.9c}$$

$$\lambda_1 \left(\sum_{k=1}^{K} P_k m_k - P_{max} M_D \right) = 0,\ k \in \{1, ..., K\}, \tag{5.10a}$$

$$\lambda_2 \left(\sum_{k=1}^{K} m_k - M_D \right) = 0, \tag{5.10b}$$

$$\lambda_{3,k} \left(\varepsilon_k - \eta \right) = 0,\ k \in \{1, ..., K\}, \tag{5.10c}$$

$$\lambda_{4,k} \left(\varepsilon_k - \varepsilon_{max,k} \right) = 0,\ k \in \{1, ..., K\}, \tag{5.10d}$$

where $H_k = \frac{h_k}{\sigma^2}$. (5.9a)-(5.9c) and (5.10a)-(5.10d) represents the stationarity and complementary slackness conditions, respectively, and $\lambda_{i,k} \geq 0$ is the Lagrange multiplier associated with the i^{th} constraint of the k^{th} VUE, $i \in \{1, 2, 3, 4\}$ corresponding to constraints (5.5b)-(5.5e), respectively. As the $\mathcal{OPBP}$ is a convex optimization problem as per Lemma 1, which satisfies the slater's constraint qualification condition, the KKT conditions are sufficient conditions for an optimal solution of the $\mathcal{OPBP}$. Therefore, the optimal solution for block lengths $\boldsymbol{m} = \{m_1, \cdots, m_K\}$, and transmit powers $\boldsymbol{P} = \{P_1, \cdots, P_K\}$ can be readily obtained by using standard convex solvers.

Since the block length solution obtained from the relaxed integer problem may violate the original integer constraint (5.4e), a low-complexity greedy search methodology is adopted to convert the positive continuous block length values to integer solution [Ren et al., 2020b]. The $\mathcal{JOA}$ for solving $\mathcal{MMDEP}$ problem is summarized in Algorithm 5. The $\mathcal{JOA}$ starts by solving the convex $\mathcal{OPBP}$ problem (5.5) to generate the non-integer block length vector denoted by $\boldsymbol{m}^o = [m_1^o, \cdots, m_K^o]$ (Line 1). The integer solution for block length vector is then initialized to $\tilde{\boldsymbol{m}}_{\boldsymbol{k}} = [\tilde{m}_1, \cdots, \tilde{m}_K] = \lfloor m_1^o, \cdots, m_K^o \rfloor$, where $\lfloor \cdot \rfloor$ indicates the floor function (Line 2). Then, there are $M_{\text{UA}} = \sum_{k=1}^{K} m_k^o - \sum_{k=1}^{K} \tilde{m}_k$ unassigned block lengths. Since the decoding-error probability is a decreasing function of the block length m_k, the $\mathcal{JOA}$ allocates one unassigned symbol to the VUE, which results in the largest decrement of the decoding-error probability, i.e., $\arg\min_{k \in K} \left(\varepsilon_k(\tilde{m}_k + 1) - \varepsilon_k(\tilde{m}_k) \right), \forall k \in \{1, ..., K\}$. This process is repeated till all unassigned block lengths are allocated (Lines $4 - 8$). The $\mathcal{JOA}$ terminates after updating the final transmit power solution $\tilde{\boldsymbol{P}}$ based on

the integer block length solution $\tilde{\boldsymbol{m}}$ obtained via the greedy search method (Line 9).

Algorithm 5: $\mathcal{JOA}$ for Solving $\mathcal{MMDEP}$ Problem

1: solve the convex problem (5.5) to generate $\boldsymbol{m}^o = [m_1^o, \cdots, m_K^o]$.
2: obtain initial integer block length solution $\tilde{\boldsymbol{m}}_{\boldsymbol{k}} = \lfloor m_1^o, \cdots, m_K^o \rfloor$.
3: determine unassigned block lengths $M_{\mathrm{UA}} = \sum_{k=1}^{K} m_k^o - \sum_{k=1}^{K} \tilde{m}_k$.
4: **while** $M_{\mathrm{UA}} > 0$ **do**
5: $\quad k^\star \leftarrow \arg\min_{k \in K} \left(\varepsilon_k(\tilde{m}_k + 1) - \varepsilon_k(\tilde{m}_k)\right)$,
6: $\quad \tilde{m}_{k^\star} \leftarrow \tilde{m}_{k^\star} + 1$,
7: $\quad M_{\mathrm{UA}} \leftarrow M_{\mathrm{UA}} - 1$,
8: **end while**
9: obtain $\tilde{\boldsymbol{P}}$ using $\tilde{\boldsymbol{m}}$.

The primary computational complexity of the proposed $\mathcal{JOA}$ comes from the concurrent solution of the block length and power allocation vectors with a computation cost of $\mathcal{O}\left((K^2+1)\right)$ for K VUEs within the network [Tseng, 2001]. Additionally, the integer conversion of block length variable via the proposed greedy search method necessitates the iterative allocation of M_{UA} unassigned symbols to the K VUE, with a computation cost of $\mathcal{O}\left(M_{\mathrm{UA}}\right)$. Thus, the overall computational complexity is $\mathcal{O}\left((K^2+1) + M_{\mathrm{UA}}\right)$.

To ensure optimality, it is crucial to determine the transmit power and block length at the beginning of each time frame. However, ensuring timely resource allocation in URLLC V2X environments using the $\mathcal{JOA}$ presents three primary challenges. First, the time complexity of $\mathcal{JOA}$ grows quadratically with network size, rendering it impractical for large-scale networks where solutions quickly become outdated due to short coherence time intervals. Second, $\mathcal{JOA}$ relies on instantaneous global CSI, which may be challenging to acquire in dynamically changing vehicular networks. Third, $\mathcal{JOA}$ considers a myopic view of the V2X environment, focusing on optimizing individual time slots without considering historical network state information. This approach may lead to sub-optimal solutions and poor per-

formance.

To address these challenges, an RL-based strategy offers long-term benefits by modeling the optimization problem as a sequential decision-making process. Using an RL-based framework, the network's historical wireless data can effectively optimize current choices in real-time, a capability absent in conventional optimization-based solutions like the $\mathcal{JOA}$. Next, we propose a robust event-triggered DRL-based strategy to solve the $\mathcal{MMDEP}$.

5.5 Event-Triggered DRL-Based Power and block length allocation Scheme

In this section, we propose a DRL-based design to solve the $\mathcal{MMDEP}$ problem. To build up the foundation for the proposed DRL-based strategy, a brief overview of the two related DRL methods, Deep Q-network (DQN) and Deep deterministic policy gradient (DDPG), is first provided, followed by the description of the state, action, reward function, for transforming the joint resource allocation problem into an RL framework. An event-triggered mechanism is introduced based on input state similarity to determine whether to trigger the DRL process or not. Finally, the event-triggered DRL-based algorithm for the optimization of transmit power and block length is proposed.

5.5.1 Deep Reinforcement Learning Overview

Reinforcement learning is a decision-making process where an agent learns the outcome of its actions over time through repeated trial-and-error interactions with its environment [Sutton and Barto, 2018]. Let $\mathcal{S}$ denote a set of possible states and $\mathcal{A}$ indicate the set of possible actions. In each time frame, the agent interacts with the environment, observes the current state $s^{(t)} \in \mathcal{S}$, takes action $a^{(t)} \in \mathcal{A}$ according to the current policy $\pi\left(s^{(t)}, a^{(t)}\right)$, where $\pi\left(s^{(t)}, a^{(t)}\right)$: $\mathcal{S} \mapsto \mathcal{A}$ is the probability of taking action $a^{(t)}$ given the current state $s^{(t)}$, receives an immediate reward $r^{(t)}$ and thereafter transitions to a next state $s^{(t+1)}$. This process is repeated, and in each time frame, the agent collects the tuple $e^{(t)} = \left(s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)}\right)$ as an experience.

The objective of the RL-based system design is to find an optimal policy that maximizes the expected return, where the return $G^{(t)}$ is defined as the cumulative total discounted reward from an initial state $s^{(t)}$ and is mathematically expressed as

$$G^{(t)} = \sum_{i=0}^{\infty} \gamma^i r^{(t+i)} \qquad 0 \leq \gamma \leq 1, \tag{5.11}$$

where the discount factor γ describes the impact of future rewards relative to those in the immediate future. Under a certain policy $\pi\left(s^{(t)}, a^{(t)}\right)$, the associated Q-function value is defined as the total expected reward of executing action $a^{(t)}$ under state $s^{(t)}$, i.e.,

$$Q^{\pi}\left(s^{(t)}, a^{(t)}\right) = \mathbb{E}_{\pi}\left[G^{(t)} \mid s^{(t)}, a^{(t)}\right]. \tag{5.12}$$

The optimal Q-function $Q^{\pi^*}\left(s^{(t)}, a^{(t)}\right)$ associated with the optimal policy $\pi^*\left(s^{(t)}, a^{(t)}\right)$ can be obtained by solving the Bellman equation recursively [Sutton and Barto, 2018]. However, obtaining the optimal Q-function is challenging, particularly with the large dimensional state and action space. To overcome this curse of dimensionality, a DQN denoted as $Q^{\pi}\left(s^{(t)}, a^{(t)}; \theta^{(t)}\right)$ parameterized by $\theta^{(t)}$, is trained to approximate $Q^{\pi}\left(s^{(t)}, a^{(t)}\right)$ from which the corresponding optimal policy $\pi^*\left(s^{(t)}, a^{(t)}\right)$ can be extracted. Additionally, to stabilize the learning, two DQNs are utilized during the training process: an evaluation network with weight parameters $\theta^{(t)}$ and a target network with parameters $\bar{\theta^{(t)}}$. The target network parameters are periodically updated by copying the weights from the evaluation network. The DRL agent continuously collects new experiences and stores them in the experience replay memory $\mathcal{M}$, which works in a *first-in-first-out* (FIFO) manner. Then, by sampling a mini-batch $\mathcal{M}^{mini}$ of experiences with length $\|B\|$ ($B = \{e_1, e_2, \cdots, e_B\}$) from $\mathcal{M}$, the DRL agent can update $\theta^{(t)}$ by adopting a proper optimizer to minimize the following mean-squared loss function

$$L\left(\theta^{(t)}\right) = \frac{1}{\|B\|} \sum_{e \in B} \left\{ \left(y^{(t)}(s^{(t+1)}, r^{(t+1)}) - Q^{\pi}(s^{(t)}, a^{(t)}; \theta^{(t)}) \right)^2 \right\}, \tag{5.13}$$

where $y^{(t)}\left(s^{(t+1)}, r^{(t)}\right) = \left\{ r^{(t)} + \gamma \max_{a^{(t+1)}} Q^{\pi}\left(s^{(t+1)}, a^{(t+1)}; \bar{\theta^{(t)}}\right) \right\}$ is the target Q-value generated by the target network. Updating the DQN weights by using batches

of randomly selected states from the experience replay memory rather than just the latest state breaks the temporal correlation among states, ultimately improving performance.

Deep-Q learning is well suited for discrete state and action spaces but cannot learn continuous stochastic policies. When the network has continuous state and action spaces, as is the case for our power allocation problem, the DDPG algorithm is a suitable alternative that can effectively deal with continuous optimization space. The DDPG algorithm works in an actor-critic manner: The evaluation actor network represents the parameterized policy network $\mu\left(s^{(t)};\psi^{(t)}\right)$ with weights $\psi^{(t)}$, and the action is extracted via the deterministic policy as $a^{(t)} = \left[\mu\left(s^{(t)};\psi^{(t)}\right) + \mathcal{W}\right]_0^{P_{\max}}$, where $\mathcal{W}$ represents the stochastic noise following a normal distribution. On the other hand, the evaluation critic network $Q^c\left(s^{(t)}, a^{(t)};\phi^{(t)}\right)$ with weights $\phi^{(t)}$ represents the value-function network, which takes the current state $s^{(t)}$ and the deterministic action $a^{(t)}$ as inputs and outputs a single Q-value to estimate the long-term reward. Similar to the DQN, target actor network $\mu\left(s^{(t)};\bar{\psi^{(t)}}\right)$ with weights $\bar{\psi^{(t)}}$, and target critic network $Q^c\left(s^{(t)}, a^{(t)};\bar{\phi^{(t)}}\right)$, with weights $\bar{\phi^{(t)}}$ are used to stabilize the learning process. The weight parameters $\phi^{(t)}$ of the critic network can be adjusted in a fashion similar to the DQN parameter update by minimizing the following critic loss function:

$$L(\phi^{(t)}) = \frac{1}{\|B\|}\sum_{e\in B}\left\{\left(y^{(t)}\left(s^{(t+1)}, r^{(t)}\right) - Q^c(s^{(t)}, a^{(t)};\phi^{(t)})\right)^2\right\}, \tag{5.14}$$

where $y^{(t)}\left(s^{(t+1)}, r^{(t)}\right) = \left\{r^{(t)} + \gamma \max_{a^{(t+1)}} Q^c\left(s^{(t+1)}, \mu(s^{(t+1)};\bar{\psi^{(t)}});\bar{\phi^{(t)}}\right)\right\}$ is the target critic network.

The actor network parameters $\psi^{(t)}$ can be updated by the gradient of the critic network output w.r.t. the action $a^{(t)}$, multiplied by the gradient of the actor network output w.r.t. its parameters $\psi^{(t)}$, and averaged over a mini-batch as follows:

$$\nabla_{\psi^{(t)}} J\left(\psi^{(t)}\right) = \frac{1}{\|B\|}\sum_{e\in B}\left\{\nabla_{\psi^{(t)}}\mu\left(s^{(t)};\psi^{(t)}\right)\nabla_{a^{(t)}}Q^c\left(s^{(t)}, a^{(t)};\phi^{(t)}\right)\right\}, \tag{5.15}$$

where $\nabla_{\psi^{(t)}}$ denotes the gradient w.r.t. $\psi^{(t)}$ and $J\left(\psi^{(t)}\right)$ is the policy objective function to be maximized. Then, based on the stochastic gradient descent technique,

the weights of the evaluation DQN, the evaluation critic network, and the evaluation actor network can be updated as

$$\begin{aligned}
\theta^{(t)} &\leftarrow \theta^{(t)} - \eta \nabla_{\theta^{(t)}} L\left(\theta^{(t)}\right), \\
\phi^{(t)} &\leftarrow \phi^{(t)} - \delta \nabla_{\phi^{(t)}} L\left(\phi^{(t)}\right), \\
\psi^{(t)} &\leftarrow \psi^{(t)} - \zeta \nabla_{\psi^{(t)}} J\left(\psi^{(t)}\right),
\end{aligned} \tag{5.16}$$

where η, δ, ζ are the learning rates of the evaluation DQN, evaluation critic network, and the evaluation actor network, respectively. The parameters of the target actor network and the target critic network are softly updated using $\bar{\psi^{(t)}} \leftarrow \tau\psi^{(t)} + (1-\tau)\bar{\psi^{(t)}}$, $\bar{\phi^{(t)}} \leftarrow \tau\phi^{(t)} + (1-\tau)\bar{\phi^{(t)}}$ to slowly track the learned evaluation network parameters, where $\tau \ll 1$ [Sutton and Barto, 2018], [Mnih et al., 2015].

5.5.2 DRL-Based Resource Allocation Framework

In this section, we cast the $\mathcal{MMDEP}$ problem (5.4) as a sequential decision-making process by mapping the key elements from the RL framework to the $\mathcal{MMDEP}$ problem. Our proposed two-layered DQN and DDPG-based framework is elaborated first, followed by the derived event-triggered DRL-based algorithm in the following section.

The $\mathcal{MMDEP}$ problem inherently constitutes a hybrid discrete-continuous action space. The two-layered network structure at the RSU optimizes the discrete block length allocation policy and the continuous transmit power allocation policy simultaneously, as shown in Fig. 5.2. For the first layer responsible for the discrete block length assignment, we propose a multi-DQN structure. The key idea is to train a separate DQN for each block length allocation decision so that the total number of DQN structures equals the number of VUEs. In this way, the scale of the block length action space is reduced from $(M_D)^K$ to $K\text{x}M_D$. On the other hand, due to the continuous optimization space of the transmit power, we utilize the DDPG-based algorithm to assign the downlink continuous power in the second layer. This approach effectively addresses the challenge of quantization errors and the consequent performance degradation resulting from the discretization of continuous actions [Yuan et al., 2021]. This two-layered structure is beneficial as it eases the

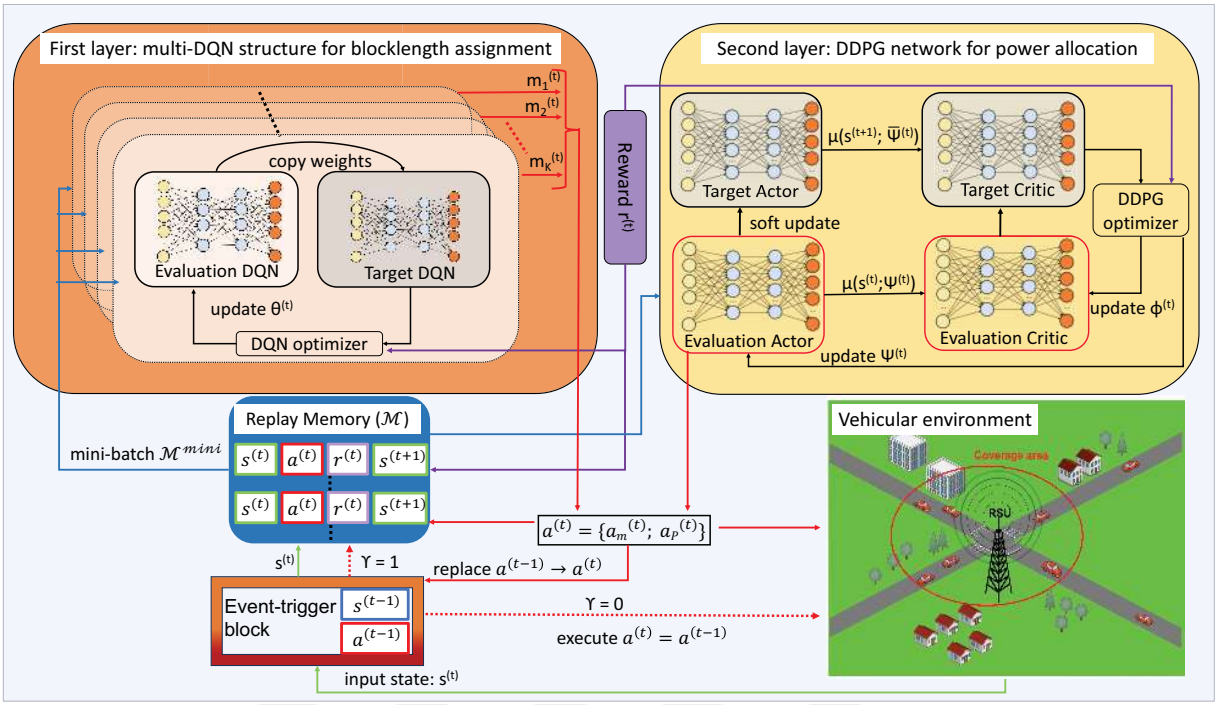


Figure 5.2: Proposed event-triggered DRL framework for power control and block length allocation.

implementation and enables the concurrent optimization of the two policies during the training phase for faster convergence.

Next, we introduce the RL framework in detail in terms of the state space, action space, and the reward function.

- *State Space* : Considering the channel correlation between adjacent frames, we utilize information from historical frames in addition to the instantaneous channel information, which offers essential data that can be used to optimize the resource allocation policies [Nasir and Guo, 2019].

  1. The historical data comprises the downlink channel gains, $\mathbf{h}^{(t-1)} = \left[h_1^{(t-1)}, h_2^{(t-1)}, \ldots, h_K^{(t-1)}\right]$, previously allocated transmit powers, $\mathbf{P}^{(t-1)} = \left[P_1^{(t-1)}, P_2^{(t-1)}, \ldots, P_K^{(t-1)}\right]$, and previously assigned block lengths, $\mathbf{m}^{(t-1)} = \left[m_1^{(t-1)}, m_2^{(t-1)}, \ldots, m_K^{(t-1)}\right]$.

2. The instantaneous information includes the channel gains, $\mathbf{h}^{(t)} = \left[h_1^{(t)}, h_2^{(t)}, \ldots, h_K^{(t)}\right]$, in the current time frame.

$$s^{(t)} = \left\{ \underbrace{\mathbf{h}^{(t-1)}, \mathbf{P}^{(t-1)}, \mathbf{m}^{(t-1)}}_{\text{historical information}}, \underbrace{\mathbf{h}^{(t)}}_{\text{instantaneous information}} \right\}. \tag{5.17}$$

The cardinality of the state space is $(4 \times K)$, depending on the number of VUEs.

- *Action Space* : The DRL agent performs two actions in each time frame (t): adjusting the block length, denoted as $a_m^{(t)}$, and setting the transmit power, denoted as $a_P^{(t)}$. The combined action of the DRL agent can be expressed as

$$a^{(t)} = \left\{a_m^{(t)}, a_P^{(t)}\right\} = \left\{ m_k^{(t)}, P_k^{(t)} |\ k = 1, \ldots, K \right\}, \tag{5.18}$$

where the discrete value $m_k^{(t)} \in \{0, 1, \ldots, M_D\}$ and the continuous value $P_k^{(t)} \in [0, P_{\max}]$ represent the block length and transmit power allocated to the k^{th} VUE at time frame t, respectively.

- *Reward function* : The flexibility of RL reward design makes it particularly appealing for solving problems with difficult-to-optimize objectives. To achieve the global objective (5.4a), while considering the stringent latency and reliability requirements, we design a reward function, which is expressed as

$$\begin{aligned} r^{(t)} = \Bigg\{ & -\alpha_1 \max_{k \in K} \left(\varepsilon_1^{(t)}, \ldots, \varepsilon_k^{(t)}, \ldots, \varepsilon_K^{(t)}\right) \\ & -\alpha_2 \max\left(\sum_{k \in \mathcal{K}} m_k^{(t)} P_k^{(t)} - P_{max} M_D, 0\right) \\ & -\alpha_3 \sum_{k \in \mathcal{K}} \left(\max\left(\epsilon_k^{(t)} - \epsilon_{th}, 0\right)\right) \\ & -\alpha_4 \max\left(\sum_{k \in \mathcal{K}} m_k^{(t)} - M_D, 0\right) \Bigg\}, \end{aligned} \tag{5.19}$$

where $\alpha_i, i \in \{1, 2, 3, 4\}$, are the positive weights to balance the utility (obtained from the objective function) and the cost in terms of constraint violation. To align the reward function with the desired objective, we include the first term in Eq.(5.19) as the negative reward to be consistent with the objective in our system model. Additionally, the unsatisfied constraints of transmit power, reliability, and latency are incorporated into the reward as penalties. The weight coefficients are carefully selected as they impact the learning efficiency and convergence of the algorithm. In practice, the coefficients α_i are hyperparameters and need to be tuned empirically. In our training, we set $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = \frac{1}{(P_{max} M_D)}$ and $\alpha_4 = \frac{1}{M_D}$. This hyperparameter selection approach balances the contribution of transmit power, reliability, and latency constraints to the desired objective, and enables stable updates of the Q-values for improved convergence.

5.5.3 Event-triggered learning

In general, RL is studied by means of Markov decision processes (MDPs), where learning is performed in a time-triggered mechanism. The periodic training for learning may be unnecessary when the environmental fluctuation is negligible. Besides, training requires high-performance computing to reduce time costs, while the required hardware is expensive and power-hungry. Inspired by event-triggered control systems [Solowjow and Trimpe, 2020, Lu et al., 2023], which activate computation or communications only when the system state deviates from the expected accuracy, we incorporate event-triggered learning into our proposed DRL framework to reduce computational cost and improve learning efficiency.

According to (5.1), for low Doppler frequency, the input states of the training agent, which includes the channel gains in consecutive time frames, are very similar due to the temporal correlation between the current and past channel conditions. This implies that the resource allocation decisions are likely to be the same in consecutive time frames. As shown in Fig. 5.2, based on the temporal correlation between the states of the agent, we introduce an event-trigger block before the initiation of

the DRL process, which checks whether to initiate the learning process for new action selection or use the former action for resource allocation. The event-triggering block holds the previous state-action pair $\left\{s^{(t-1)}, a^{(t-1)}\right\}$, and compares the former state of the agent against the current observed state. If the absolute difference is below the event-triggering threshold, then the former action is executed without initiating the learning process, thus reducing the use of computational resources over time frame t. Specifically, the triggering criterion is defined as follows:

$$\Upsilon = \begin{cases} 1, & \text{if } \left\|s^{(t)} - s^{(t-1)}\right\| \geq \upsilon \\ 0, & \text{otherwise} \end{cases} \tag{5.20}$$

where $\upsilon \geq 0$ is the trigger threshold, $\Upsilon = 1$ means initiating the DRL process, and $\Upsilon = 0$ means using former state-action pair.

5.5.4 Proposed DRL-based Resource Allocation Algorithm

In this section, we present our proposed event-triggered DRL algorithm for solving the $\mathcal{MMDEP}$ problem. The proposed event-trigger learning and DRL-based algorithm for the joint block length assignment and transmit power allocation is denoted by Event-triggered DDPG-DQN, and is summarized in Algorithm 6, which includes three phases, i.e., initialization, random experience collection, and event-triggered training and execution.

In the *initialization* phase, K evaluation DQNs $Q\left(s^{(t)}, a^{(t)}; \theta^{(t)}\right)$, and K corresponding target DQNs $Q\left(s^{(t)}, a^{(t)}; \theta^{\bar{(t)}}\right)$ are constructed as a first layer at the RSU, meanwhile evaluation actor network $\mu\left(s^{(t)}; \psi^{(t)}\right)$, target actor network $\mu\left(s^{(t)}; \psi^{\bar{(t)}}\right)$, evaluation critic network $Q^c\left(s^{(t)}, a^{(t)}; \phi^{(t)}\right)$, and the corresponding target critic network $Q^c\left(s^{(t)}, a^{(t)}; \phi^{\bar{(t)}}\right)$ are established as a second layer at the RSU. The weights $\theta^{(t)}, \psi^{(t)}, \phi^{(t)}$ of the evaluation networks are randomly initialized, whereas the target network weights $\theta^{\bar{(t)}}, \psi^{\bar{(t)}}, \phi^{\bar{(t)}}$ are initialized with the weights of the corresponding evaluation network weights, i.e., $\theta^{(t)} = \theta^{\bar{(t)}}$, $\psi^{(t)} = \psi^{\bar{(t)}}$, and $\phi^{(t)} = \phi^{\bar{(t)}}$, respectively (Lines $1 - 3$).

Algorithm 6: Proposed Event-triggered DDPG-DQN Algorithm

Initialization:

Initialize the replay memory block, $\mathcal{M}$, and the event-trigger block with $s^{(t-1)} = 0,\ a^{(t-1)} = 0$;

Randomly initialize $\theta^{(t)}$, $\psi^{(t)}$, $\phi^{(t)}$;

Initialize the target network weight parameters $\theta^{\bar{(t)}} \leftarrow \theta^{(t)}, \psi^{\bar{(t)}} \leftarrow \psi^{(t)}, \phi^{\bar{(t)}} \leftarrow \phi^{(t)}$;

Random experience collection:

for $t = 0, 1, \ldots T_{rand}$ **do**

  Execute the random joint action $a^{(t)}$;

  Obtain the reward using (5.19);

  Observe the new state $s^{(t+1)}$;

  Store experience $e^{(t)} = \left(s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)}\right)$ in $\mathcal{M}$;

  if $t > B$ **then**

    Sample B experiences from $\mathcal{M}$;

    Update the weights of evaluation networks $\theta^{(t)}$, $\psi^{(t)}$ and $\phi^{(t)}$ with (5.16);

  end

end

Event-triggered training:

for $t = T_{rand}, T_{rand}+1, \ldots \mathcal{T}_{train}$ **do**

  Compute event-triggering condition Υ according to (5.20);

  if $\Upsilon = 1$ **then**

    Choose the block length action $a_m^{(t)}$ following the $\epsilon-$greedy policy;

    Generate the deterministic power action $a_P^{(t)} = \left[\mu\left(s^{(t)}; \psi^{(t)}\right) + \mathcal{W}\right]_0^{P_{\max}}$;

    $a^{(t)} = \left\{a_m^{(t)}, a_P^{(t)}\right\}$;

  end

  else

    Execute the previous action $a^{(t-1)}$;

  end

  Evaluate the reward using (5.19), next state $s^{(t+1)}$ and store experience $e^{(t)} = \left(s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)}\right)$ in $\mathcal{M}$;

  Randomly sample B experiences from $\mathcal{M}$;

  Update the weights of evaluation networks $\theta^{(t)}$, $\psi^{(t)}$ and $\phi^{(t)}$ with (5.16);

end

In the *random experience collection*, we let the agent do the random walk for T_{rand} time frames, which induces changes in the channel conditions and consequently allows the policy to observe more variant states during its training. This procedure intuitively increases the algorithm's robustness to the changes in channel conditions. At the beginning of each time frame, the RSU receives the channel gains from all VUEs to establish the current state $s^{(t)}$. The DRL agent selects a random transmit power and block length action, obtains the reward using (5.19), and transitions to the next state $s^{(t+1)}$ due to the channel variation in the next time frame. Meanwhile, the experience $e^{(t)} = \left(s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)}\right)$ is stored in the experience replay memory $\mathcal{M}$ (Lines $5-8$). Note that the replay buffer memory capacity $\mathcal{M}$ is much larger than the batch size B, and RSU begins to sample from $\mathcal{M}$ when at least B sample experiences are available. A mini-batch of experiences of size B from $\mathcal{M}$ is used to train the evaluation DQNs, the actor network, and the critic network, i.e., to minimize the DQN loss in (5.13) and the critic loss function in (5.14). Afterwards, the agent updates the actor policy by the sampled policy gradient with (5.15), the evaluation network parameters with (5.16), and soft updates the target network parameters with those of the evaluation networks (Lines $10-11$).

In the *event-triggered training* phase ($t \geq T_{rand}$), at the beginning of time frame t, the RSU receives the channel gains from all VUEs. The current state of the agent $s^{(t)}$ is passed to the trigger block to compute (5.20) (Line 15). The DRL process is initiated if $\Upsilon = 1$. The current state is sent to the multi-DQN structure (first layer) and the DDPG-based network (second layer) for block length and transmit power estimation, respectively. Specifically, the DRL agent outputs the block length allocation action $a_m^{(t)}$ using K separate DQNs while following the $\epsilon-$greedy policy, and the deterministic power action $a_P^{(t)} = \left[\mu\left(s^{(t)}; \psi^{(t)}\right) + \mathcal{W}\right]_0^{P_{\max}}$ using the actor network (Lines $16-18$). Otherwise, the agent executes the previous action stored in the event-trigger block (Lines $19-22$). The agent obtains the reward using (5.19), the next state $s^{(t+1)}$, and stores the current experience tuple $e^{(t)}$ in the replay buffer memory $\mathcal{M}$ (Line 24). Then, by sampling a mini-batch of B experiences, the agent trains the evaluation DQNs, the actor network, and the critic network in a manner

similar to that described in the random experience collection phase (Lines $25-26$). Finally, upon convergence, the algorithm outputs the trained parameters; $\widetilde{\theta}^{(t)}$, $\widetilde{\psi}^{(t)}$, $\widetilde{\phi}^{(t)}$ for the evaluation DQNs, the evaluation actor, and the evaluation critic network, respectively.

The major computational complexity of Algorithm 6 comes from training K evaluation DQNs, and four DNNs in the actor-critic network structure. Assuming that the evaluation actor network, the critic network, and each DQN uses J, N, and L fully connected layers, respectively, the computational complexity can be calculated using $O\Big(\sum_{j=0}^{J-1} u_j u_{j+1} + \sum_{n=0}^{N-1} u_n u_{n+1} + \sum_{l=0}^{L-1} u_l u_{l+1}\Big)$, where u_i denotes the number of units in i-th fully connected layer, and u_0 indicates the input size.

5.6 Performance Evaluation

In this section, we evaluate the performance of our proposed event-triggered DDPG-DQN based approach in comparison to the three state-of-the-art benchmark algorithms, namely, our proposed joint optimization algorithm ($\mathcal{JOA}$), the single-DQN scheme, the DDPG-DQN scheme, and the random allocation algorithm. The single-DQN scheme considers a single DQN for the joint transmit power and block length allocation by quantizing the power into ten discrete levels. This scheme is only available for discrete action spaces due to the inherent limitation of the DQN and is based on the adaptation of the joint DRL-based methodology proposed in [Tan et al., 2021]. The DDPG-DQN scheme optimizes the continuous transmit power and block length in each time frame by utilizing the proposed DRL-based framework without the event-triggered learning mechanism. Finally, the random allocation algorithm chooses a random transmit power and block length action at the beginning of every time frame.

To compare the performance of different learning schemes, the results are averaged over ten independent simulation runs with random initialization. The simulation setup constitutes two main phases, i.e., the training phase and the testing phase. The training phase lasts $\mathcal{T}_{train} = 20,000$ time frames, whereas the subsequent $5,000$ time frames are dedicated to the testing phase, where the well-trained DNNs

Table 5.1: Simulation parameters

Parameter	**Value**
Maximum transmit power P_{max}	23 dBm
Noise power σ^2	-114 dBm
Vehicle receiver noise figure	9 dB
Vehicle speed v_s	60 km/hr
Vehicle placement model	spatial Poisson
Lane width	4 m
Carrier frequency f_c	2 GHz
Shadowing standard deviation σ_{sh}	3 dB
Radius	500 m
RSU antenna height	25 m
RSU antenna gain	8 dBi
RSU receiver noise figure	5 dB
Distance between RSU and highway	35 m
Vehicle antenna height	1.5 m
Vehicle antenna gain	3 dBi
Payload size L	100 bits
SNR threshold γ_{th}	0 dB

are utilized to evaluate the performance of the different learning algorithms. To obtain the figures, a second-order Savitzky-Golay smoothing filter with a window size of 503 is utilized for smoothness. All simulations are performed on a 10-Core Intel(R) Xenon(R) Silver 4114, 2.2GHz system equipped with an Nvidia Quadro P2000 graphics processing unit (GPU).

5.6.1 Simulation Setup

We investigate the performance of our proposed algorithm against several schemes by simulating a vehicular network with multiple VUEs. In particular, we follow the simulation setup for a freeway model described in 3GPP TR 36.885, which describes in detail the vehicular channel models, vehicle mobility, and vehicular data traffic [3GPP, 2023]. In particular, N V2I links are initiated by N vehicles and the K V2V links are formed between each vehicle with its surrounding neighbors. The

Table 5.2: Hyperparameters for actor network

Layers	Input	$L_1^{(a)}$	$L_2^{(a)}$	$L_3^{(a)}$	Output
Neuron number	$(4K)$	500	250	120	1
Activation function	Linear	Relu	tanh	Relu	Sigmoid
Action noise $\mathcal{W}$	Standard Gaussian distribution $\mathcal{W}(0,0.1)$				
Mini-batch size B	128				
Random experience collection period T_{rand}	$3 \times B$				
Replay buffer memory capacity	500				

Table 5.3: Hyperparameters for critic network and deep-Q network

DNN type	Critic network					DQN				
Layers	Input	$L_1^{(c)}$	$L_2^{(c)}$	$L_3^{(c)}$	Output	Input	$L_1^{(DQN)}$	$L_2^{(DQN)}$	$L_3^{(DQN)}$	Output
Neuron number	$(4K)$+1	500	250	120	1	$(4K)$	500	250	120	M_D
Activation function	Linear	Relu	tanh	Relu	Linear	Linear	Relu	tanh	Relu	Linear
Optimizer	RMSPropOptimizer									
Mini-batch size B	128									
Replay buffer memory capacity	500									
Discount factor γ	0.5									
Initial learning rate $\delta^{(0)}$	0.05									
Learning decay rate α_L	0.0001									
Initial exploration rate $\epsilon^{(0)}$	0.1									
ϵ-decay rate α_ϵ	0.0001									

vehicles are distributed according to Poisson distribution on a multi-lane highway with the BS at the center and at a fixed distance from the highway. The considered highway is a 3×2 lane with a fixed lane width. The vehicle density on the highway is determined by the vehicle speed v_s, and the average inter-vehicular distance is $2.5 \times v_s$, which is considered fixed in our simulation results. The large-scale fading is modeled using the WINNER path-loss model $PL(d) = 128.1 + 37.6 \log_{10}(d)$, where d is the distance in [km] of the VUE from the RSU [Kyösti et al., 2007]. Fast fading has been modeled using Rayleigh fading with the time correlation between successive time frames, as represented by Jake's model. The important simulation parameters are provided in Table 5.1.

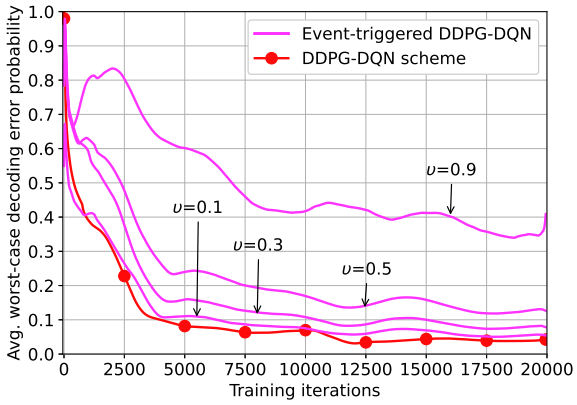
The hyperparameter values for the actor DNN, the critic DNN, and the DQN are tabulated in Table 5.2 and 5.3, respectively. The tabulated parameters are fine-

tuned to reach the desired results. The DNNs used to implement the actor network, the critic network, and the DQN have a similar architecture with one input layer, three hidden layers, and one output layer. The rectifier linear unit (ReLu) and hyperbolic tangent (tanh) functions are used alternatively as the activation function since this scheme leads to faster convergence, as per our observation. The input states are normalized and pre-processed to optimize the performance. In particular, logarithmic transformation is applied to the input states for event-triggered control and data processing before feeding them to the DNNs. Logarithmic representation is preferred since channel amplitudes often vary by order of magnitude. The output of the actor network is scaled to be between 0 and P_{max}. To encourage exploration during training, Gaussian action noise $\mathcal{W}$ with mean 0 and standard deviation of 0.05 is added to the actor network output. The replay buffer capacity and the mini-batch size are set to 500 and 128, respectively. The ϵ-greedy algorithm is initialized with $\epsilon^{(0)} = 0.1$ and the level of exploration is adaptively reduced according to $\epsilon^{(t+1)} = \max\left\{0, (1-\alpha_\epsilon)\,\epsilon^{(t)}\right\}$, where α_ϵ =0.0001 is the ϵ-decay rate.

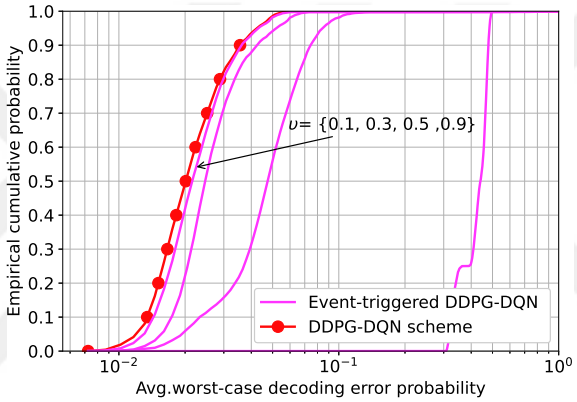
5.6.2 Analysis of Event-triggered learning

We investigate the impact of changing the trigger threshold value υ on the training convergence and number of DRL process executions of the proposed event-triggered DDPG-DQN scheme. Note that the input states are normalized and scaled before evaluating the trigger condition.

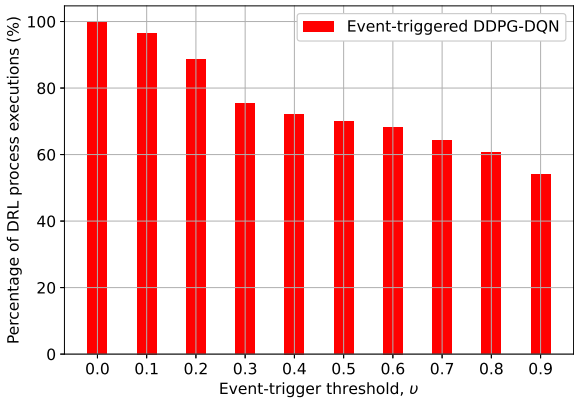
Fig. 5.3 demonstrates the impact of different event-trigger threshold values on the training convergence and complexity of the proposed algorithm in terms of DRL process executions. As shown in Fig. 5.3a and 5.3b, for low values of threshold, i.e., $\upsilon = 0.1$, the reliability performance of the proposed event-triggered DDPG-DQN scheme is very close to that of the DDPG-DQN scheme and the average performance gap between the two schemes is less than $\sim 1\%$. On the other hand, for very high threshold values, i.e., $\upsilon = 0.9$, the reliability performance significantly deteriorates in comparison to the DDPG-DQN scheme, and the event-triggered DDPG-DQN scheme achieves only $\sim 11.12\%$ of the reliability performance provided by the



(a)



(b)



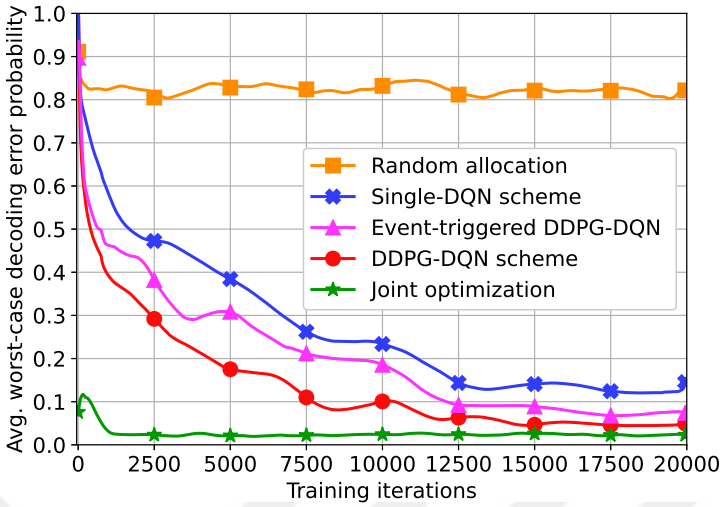
(c)

Figure 5.3: Average worst-case decoding-error probability performance of the algorithms, where (a) shows the training performance, (b) shows the testing performance for different event-trigger threshold values $\upsilon = \{0.1, 0.3, 0.5, 0.9\}$, and (c) shows the percentage of DRL executions for varying trigger threshold values, with $K = 20$, $P_{\max} = 23$ dBm, and $M_D = 300$ symbols.

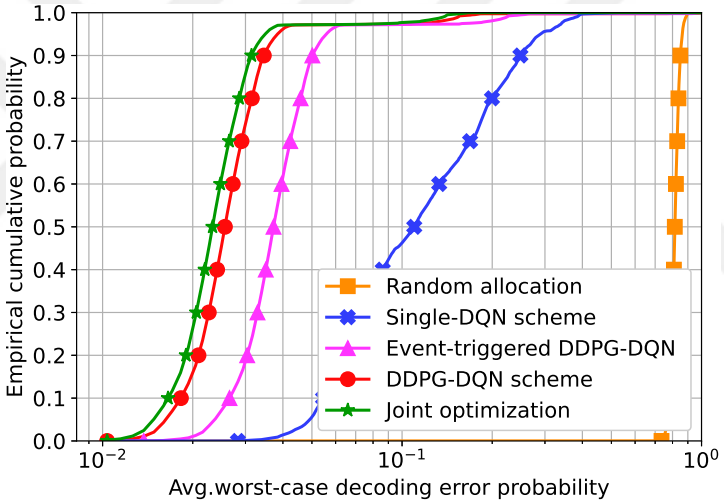
DDPG-DQN scheme. This performance degradation can be explained by the agent's reduced exposure to useful information critical for optimizing its action. Fig. 5.3c illustrates the percentage of DRL executions for varying trigger threshold values in the training phase. Increasing the threshold value results in the reduction of the number of DRL process executions. For instance, with $\upsilon = 0.9$, the DRL agent only trains the network for $\sim 50\%$ of the total training duration but provides poor reliability performance. Combining the results from Fig. 5.3a, 5.3b, and 5.3c, we can observe an interesting trade-off between the reliability performance and computation complexity in terms of the percentage of DRL executions. A moderate value of the threshold, e.g., $\upsilon = 0.3$, can achieve performance close to the DDPG-DQN scheme with an average performance gap less than $\sim 3\%$ while reducing the percentage of DRL executions by up to 24%. Therefore, for the subsequent simulation results, we fix the value of the event-trigger threshold to $\upsilon = 0.3$. Unless otherwise stated, the default parameters are set as follows: $P_{max} = 23$ dBm, $M_D = 300$, and $\upsilon = 0.3$.

5.6.3 Performance Comparison of Algorithms

Fig. 5.4a illustrates the training convergence behavior of the different algorithms. At the start of data transmissions, the average worst-case decoding error probability is identical for the proposed event-triggered DDPG-DQN algorithm and the random resource allocation algorithm since the proposed algorithm randomly assigns the power and block length to each VUE to collect experiences. From the figure, both the proposed event-triggered DDPG-DQN and the DDPG-DQN schemes converge to around 95% of the average worst-case decoding-error probability obtained using the joint optimization method. The average worst-case decoding error probability using the proposed algorithm decreases rapidly compared to the single-DQN scheme and converges after around 13,000 time frames. In contrast, the single-DQN scheme exhibits slow and poor convergence. This is mainly due to the high complexity of the action space of the single-DQN scheme, where a single DQN is used to determine the power and block length allocation actions for all VUEs. Fig. 5.4b demonstrates the effectiveness of the proposed algorithm by empirically validating the performance of



(a)



(b)

Figure 5.4: Average worst-case decoding-error probability performance of the algorithms (a) Training, (b) Testing- Empirical CDF.

the different algorithms in the testing stage. The proposed algorithm outperforms the single-DQN scheme and achieves on average 21% higher reliability compared to the single-DQN scheme on the testing data. This means that by exploiting the well-trained DNNs, the proposed algorithm can effectively learn the patterns of the environment and provide close to optimal results on the environmental states that did not appear before.

Fig. 5.5 illustrates the impact of maximum available symbols M_D on the aver-

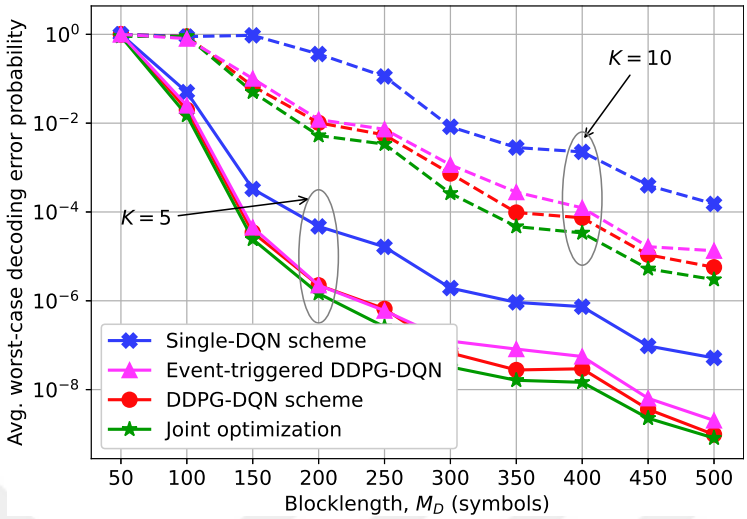


Figure 5.5: Optimized average worst-case decoding-error probability of the algorithms versus the number of symbols M_D.

age worst-case decoding-error probability. The error probability decreases with the increase in total available block length as each VUE gets a larger percentage of the total available block length resource, leading to better reliability performance. The event-triggered DDPG-DQN scheme remarkably achieves $\sim$ 95% of the reliability performance of the joint optimization scheme and $\sim$ 98% of the reliability performance offered by the DDPG-DQN algorithm, while outperforming the single-DQN for $K = 5$ and $K = 10$ VUEs. The decoding-error probability increases with the number of VUE pairs K since more VUEs are competing for the fixed available block length resource.

Fig. 5.6 shows the effect of maximum transmit power on the average worst-case decoding-error probability. Higher transmit power leads to better reliability performance, as expected. The decoding-error probability increases with the number of VUEs due to the limited power resource and degradation of the available SNR with high road traffic density. The proposed event-triggered DDPG-DQN algorithm performs close to the joint optimization scheme, and the performance gap between the event-triggered DDPG-DQN and the DDPG-DQN scheme is less than 2% for different values of P_{max}. On the other hand, the single-DQN scheme cannot keep up

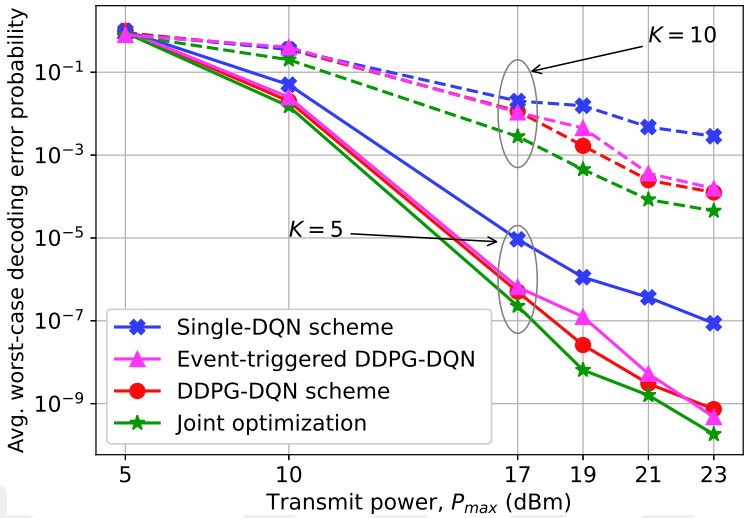


Figure 5.6: Optimized average worst-case decoding-error probability of the algorithms versus the maximum transmit power P_{max}.

with the proposed approach in terms of reliability performance. This is mainly due to the large action space of single-DQN compared to the proposed approach, which utilizes a multi-DQN structure. Besides, the quantization of power into discrete levels using the single-DQN approach discards certain pertinent information that may be crucial in determining the most suitable option for power allocation.

Fig. 5.7 provides the average reliability performance of the algorithms for different numbers of vehicles. Clearly, an increase in the number of VUEs results in a degradation of the reliability performance of all the algorithms since more vehicles are competing for the same limited resources. The random allocation strategy cannot satisfy the reliability requirement at all due to the random allocation of power and block length. The single-DQN scheme exhibits a degraded performance for all considered vehicular densities because as the vehicular network size scales up, the network topology as well as the number of possible actions increase rapidly, making it very challenging to explore the whole action space to find out the optimal action choice. On the other hand, the proposed event-triggered DDPG-DQN achieves on average 95% of the reliability performance of the joint optimization solution, and the performance gap between the event-triggered DRL scheme and the DDPG-DQN

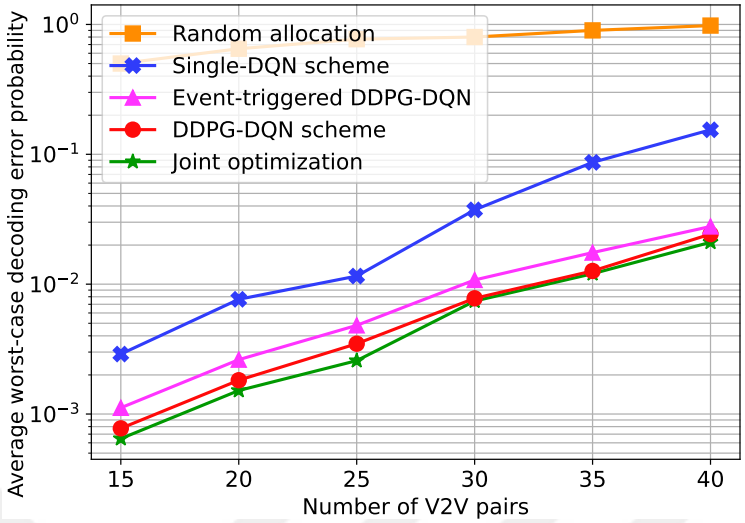


Figure 5.7: Optimized average worst-case decoding-error probability of the algorithms versus the number of VUEs K.

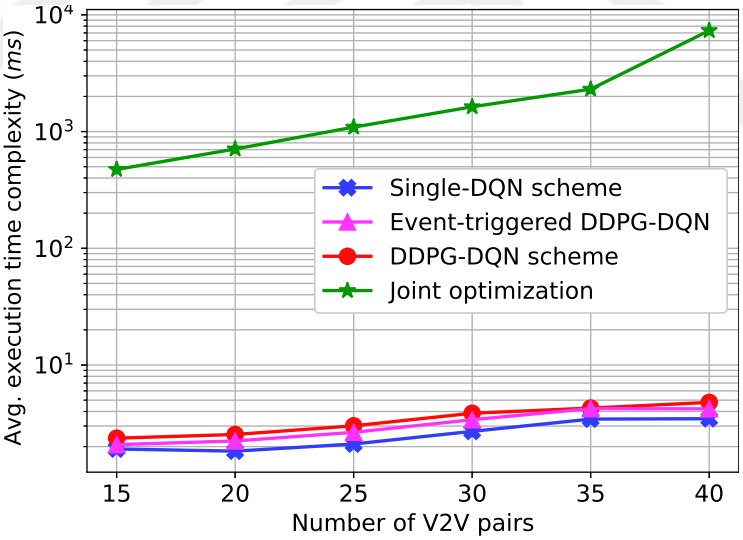


Figure 5.8: Average execution time complexity of the algorithms for different number of VUEs K.

scheme is below 5% for different vehicular densities, demonstrating the robustness of the proposed algorithm to network size.

Fig. 5.8 illustrates the average run-time of the algorithms for different network sizes. As the network size increases, the run-time complexity increases significantly

for the joint optimization algorithm due to its associated second-order polynomial growth rate in K, which limits its scalability. In contrast, once trained well, the DRL agent on average can determine its power and block length assignment action using the proposed algorithm in less than 0.8ms. Another observation is that the single-DQN scheme has the lowest run-time since a single trained network executes the two actions. However, the reduced run-time for the single-DQN scheme comes at the cost of poor convergence and degraded reliability performance as the network size increases. The proposed event-triggered DDPG-DQN scheme and the DDPG-DQN scheme utilize the same network architecture and have almost the same run-time for allocating resources for different vehicular densities. It should be pointed out that the proposed event-triggered DDPG-DQN scheme is trained using $\sim$ 75% of total training executions. Table (5.4) shows the average time complexity of various algorithms computed across training steps and different network sizes (K={15, 20, 25, 30, 35, 40} vehicles). From the table, the average time to train the proposed two layers (multiple DQNs and the actor-critic network) is around 11.2ms, which is significantly lower than that of DDPG-DQN, and the single-DQN schemes. Additionally, the average time required for calculating transmit power and block length allocation with the proposed solution is around 0.8ms, significantly less than the average calculation time of approximately 93ms needed for employing the joint optimization algorithm. Hence, the proposed framework can allocate the radio resources within the millisecond-level transmission delay requirement of real-time URLLC V2X applications. Since the computational capability of RSUs in real-world V2X networks exceeds that of the computers utilized in simulations, the average time required for training the evaluation networks and determining transmit power and block length solutions can be further reduced in practical networks.

5.6.4 Impact of Vehicular Mobility and Reliability Performance:

So far, we have presented results under fixed vehicle speed, which implies a static Doppler frequency and, therefore, a constant correlation between channels in adjacent time frames. However, in practical scenarios, VUEs may change their speed as

Table 5.4: AVERAGE TIME COMPLEXITY.

Scheme	**Average Training Time** (ms)	**Calculation with trained network** (ms)
Event-triggered DDPG-DQN	11.2	0.8
DDPG-DQN	21.7	0.8
Single-DQN	41.3	0.6

they move around, and it is essential to consider vehicle mobility's impact on DRL algorithms' performance. The convergence rate of the proposed DRL-based algorithm depends on the Doppler frequency. As we decrease Doppler frequency from $f_D = 112$ Hz ($v_s = 60\,\text{km/hr}$) to $f_D = 28$ Hz ($v_s = 15\,\text{km/hr}$), the average decoding error-probability performance remains unchanged, but the convergence time drops from 13,000 time frames to 6,000 time frames. Intuitively, this decline in convergence time with lower Doppler frequencies is related to the variability of the states observed by the agent, which is proportional to the Doppler frequency.

To evaluate the reliability performance of the trained model in the deployment (testing phase), we use the metric of *average network-wide reliability* ($\mathcal{ANR}$) defined as the average ratio of the number of vehicles with decoding error probability no greater than $\epsilon_{\max,k}$, to the total number of vehicles, formulated as:

$$\mathcal{ANR} = \frac{1}{\mathcal{T}_{test}} \sum_{t=1}^{\mathcal{T}_{test}} \sum_{k=1}^{K} \frac{\mathbb{I}\left(\epsilon_{\max,k} > \epsilon_k^{(t)}\right)}{K} \tag{5.21}$$

where $\mathbb{I}$ represents the indicator function, and the averaging is performed across the testing steps $\mathcal{T}_{test}$. In our evaluation, we consider a network with $K = 5$, $P_{\max} = 23\,\text{dBm}$, $M_D = 300$ symbols. We define two sets of error probability constraints $\varepsilon^*_{max} = [10^{-2}, 5\times10^{-2}, 10^{-3}, 5\times10^{-3}, 5\times10^{-2}]$, and $\varepsilon^{**}_{max} = [10^{-6}, 5\times10^{-5}, 10^{-4}, 5\times10^{-5}, 5\times10^{-4}]$, representing the QoS requirements for different VUEs. For ε^*_{max}, the proposed scheme achieves an $\mathcal{ANR}$ of 96% compared to 87% and 35% $\mathcal{ANR}$ of single-DQN and random allocation schemes, respectively. When considering more strict reliability requirements (ε^{**}_{max}), the $\mathcal{ANR}$ decreases for all schemes due to smaller values of the target errors impacting the network-wide reliability negatively. The proposed scheme still achieves an $\mathcal{ANR}$ of 88%, significantly higher than the

69% and 23% $\mathcal{ANR}$ of the single-DQN and random allocation schemes, respectively.

5.6.5 Discussion on Implementation and Future Extensions:

Training procedure: In the proposed event-triggered DDPG-DQN algorithm, offline training is employed to mitigate the computational overhead of learning the mapping between input states and output actions. Subsequently, during online network deployment, this mapping is executed using the pre-trained DNNs, bypassing the need for intensive computations [She et al., 2021]. Offline training can be performed using a digital twin of the vehicular network, incorporating network topology, channel models, and QoS requirements. The event-triggered DDPG-DQN algorithm can utilize this twin for offline initialization and network training at the central server. This approach allows the agent to explore in a simulated environment, mitigating real-world risks [Dong et al., 2019].

Analysis of constraint violations: Our analysis of constraint violations in the testing stage reveals that the DRL agent does not always satisfy the power, aggregate delay, and reliability constraints. However, the well-trained agent satisfies the constraints in $\sim$ 95% and $\sim$ 90% of the total testing executions for $K = 5$ and $K = 20$ VUEs, respectively, with $P_{max} = 23$ dBm, $M_D = 300$ symbols. Ensuring zero-constraint violations in real-time operations remains a significant challenge in reinforcement learning. On the one hand, providing theoretical guarantees for feasibility during online execution is crucial for DRL-based URLLC applications. On the other hand, finding the solution of constrained DRL is more challenging than standard DRL-based solutions. Constraint-guided DRL requires the agents to maximize the long-term reward while satisfying the constraints as the long-term cost. In this regard, primal-dual methods [Liang et al., 2017] and Teacher-Student learning framework [Zheng et al., 2022] offer a promising approach for constrained DRL-based solutions. The primal-dual method can be applied to optimize the constraints in the dual domain to achieve the target balance among the objective (long-term reward) and the long-term cost [She et al., 2021]. In the Teacher-Student framework, the student (DRL agent) seeks guidance from the teacher (domain-specific

optimization-based algorithm) to select an action before executing it in the wireless environment, thereby accelerating the training process and safeguarding the agent against actions that would violate specific QoS requirements.

Dynamic network topology: In practical scenarios, the number of vehicles under the coverage of RSU changes dynamically. To accommodate this variability, state encoding adjusts for changes in the number of VUEs under RSU coverage [Vinitsky et al., 2018]. If the number of VUEs exceeds the trained agent's state space dimensions, information from the extra VUEs is not included in the state. Instead, the RSU establishes the corresponding DNNs for the incoming vehicle and transfers the weights of the already-trained model to the newly established DNNs. Moreover, if one or more vehicles leave the coverage area, the agent's state is padded with zeros. It is worth pointing out that nearby vehicles often experience similar channel quality and environment observations. Therefore, the system's performance does not degrade significantly when new vehicles arrive or leave, as they can be allocated resources designated for the neighboring vehicles. Further, to avoid modifying the structure of the DNNs and retraining all the parameters with a changing number of vehicular links, graph neural networks (GNNs) are a promising avenue to explore for obtaining scalable solutions [Gu et al., 2024].

Explainable AI: For centralized learning-based algorithms, as is the case with our proposed framework, explainable AI (XAI) can be leveraged to further reduce the model's complexity and improve its robustness by explaining the learned policies during the training process [Khan et al., 2024]. XAI can alleviate the DRL model complexity and speed up the convergence time by offering methods for eliminating less important input features, aggregating identical states based on state abstraction in Markov decision processes, and utilizing model compression techniques. XAI can further improve the interpretability and robustness of DRL model decisions by providing methods for testing the system's robustness, identifying potential outliers, and verifying the credibility of the decision outcomes. Additionally, to enhance the adaptability of the proposed DRL-based solution to new environment scenarios and tasks requiring computation-intensive training, emerging schemes such

as model-agnostic meta-learning can be incorporated into the proposed framework [Yuan et al., 2021].

5.7 Conclusion

In this chapter, we have considered the problem of joint resource allocation with short packet transmission in a URLLC-enabled V2X network. To ensure fair reliability among the vehicles, we have developed an optimization framework to minimize the maximum decoding error probability among the vehicles subject to stringent latency and reliability requirements in terms of the limited block length and tolerable decoding error probability. We have proposed an optimization theory based solution strategy using the joint convexity analysis and evaluation of the KKT optimality conditions to obtain the joint solution. We have designed a computationally efficient DQN and DDPG-based framework that optimizes both the discrete block length and continuous power allocation variables simultaneously under practical URLLC constraints. Additionally, we have incorporated event-triggered learning into our proposed DRL-based methodology, resulting in an interesting trade-off between the reliability performance and computation complexity in terms of the percentage of DRL process executions. Simulation results have demonstrated that by incorporating the event-trigger mechanism, the proposed algorithm can achieve $\sim 98\%$ and $\sim 95\%$ of the reliability performance offered by the DDPG-DQN algorithm and the joint optimization scheme, respectively, while remarkably reducing the DRL executions by up to 24%.

Chapter 6

CONCLUSION

This thesis highlights the transformative potential of XAI in enhancing transparency, robustness, and efficiency in RRM for AI-native 6G networks. We demonstrate how XAI methods offer a systematic approach to interpreting decisions made by black-box AI models, enhancing decision robustness and improving algorithm performance by reducing input size, model complexity, and convergence time. Additionally, we outline the key explainability and robustness techniques, along with a design methodology for integrating these solutions into practical wireless applications. We also provide two practical case studies that illustrate the application of these techniques for model simplification and improving the robustness of RRM. The first case study introduces a robust, interpretable DL-based beam alignment framework for mmWave MIMO systems, utilizing an XAI framework to prioritize input features, reducing input size and beam alignment overhead while enhancing resilience against adversarial inputs. The second case study considers DRL-based RRM in vehicular networks, focusing on joint transmit power and spectrum allocation. It introduces a model-agnostic XAI methodology to interpret the inference process of multiple trained DRL agents. A novel SHAP-driven feature importance ranking algorithm is proposed to optimize the DRL agents' input state size. This is accomplished through an automated feature selection process that eliminates low-importance features, resulting in a simplified model with fewer network parameters, reduced training time, and reasonable performance compared to the original model. In both case studies, the explainability pipelines developed to enhance model simplification and robustness are based on post-hoc XAI techniques and are designed to be model-agnostic.

Additionally, we have considered the problem of joint resource allocation with

short packet transmission in a URLLC-enabled vehicular network. To ensure fair reliability among the communicating vehicles, we have developed an optimization framework to minimize the maximum decoding error probability among the vehicles subject to stringent latency and reliability requirements in terms of the limited block length and tolerable decoding error probability. We have designed a computationally efficient DRL-based framework that optimizes both the discrete block length and continuous power allocation variables simultaneously under practical URLLC constraints. Additionally, we have incorporated event-triggered learning into our proposed DRL-based methodology, resulting in an interesting trade-off between the reliability performance and computation complexity in terms of the percentage of DRL process executions. The results demonstrate a noteworthy reduction in training and runtime complexity compared to the conventional optimization-based solutions.

For the future work of this thesis, the following research directions are proposed for advancing XAI-based RRM solutions:

- **Transferability of the Post-Hoc Explainability**: While transfer learning-based approaches have shown good performance in training based on small datasets using pre-trained models for similar RRM problems, the transfer of knowledge (criterion for selection of layers/ learned weights) from the pre-trained source model to build the target model has not been thoroughly explored. Leveraging insights obtained through XAI can assist in reapplying the solution to several AI-empowered applications. Post-hoc explainability techniques that simplify models are usually closely linked to the specific machine learning model (ML) model and network architecture used. There is a need for generalized XAI techniques that can be applied to a variety of RRM-related AI/ML systems and applications, as well as evaluation metrics to quantify how faithfully a given explanation mimics the behavior of the underlying model.

- **Twin Systems for Performance and Reasoning:** There is a need for intelligent XAI-enabled systems that can capture the current wireless environment for decision-making while also being capable of reasoning and executing au-

tonomous decisions. AI-empowered network architectures need to be designed to incorporate explainability into wireless resource management decisions in the near future, which can be achieved by developing a parallel/twin XAI system working alongside the primary AI system. This low-complexity XAI-based twin model can then replace the primary AI system over time to enhance the performance and trustworthiness of RRM algorithms.

- **Real-Life Assessments:** Since XAI methods currently envisioned for RRM problems are evaluated in simulated or controlled environments, their performance may not reflect their efficacy in real-world environments. XAI-based models must be rigorously tested in practical wireless environments to facilitate real-time validation of different AI-based RRM solutions. Additionally, existing XAI-empowered algorithms lack automation in their inference process and suffer from performance-explainability trade-off. Striking the right balance between explainability versus performance and minimizing manual intervention is a cornerstone for driving autonomous network functions.

- **XAI integration in Open Radio Access Networks (O-RAN):** AI/ML algorithms in O-RAN framework are implemented as external applications, i.e., xApps, deployed on the near real-time RAN Intelligent Controller. To reveal AI decision logic and harness insights to optimize specific components of the AI lifecycle in O-RAN environments, it is important to integrate XAI techniques into the existing architectures. In this regard, XAI twin systems can act as a fundamental enabler for O-RAN environments in future 6G wireless networks.

BIBLIOGRAPHY

[3GPP, 2023] 3GPP (2023). Overall description of radio access network (RAN) aspects for vehicle-to-everything (V2X) based on LTE and NR. Tech. Rep. 37.985 V18.0.0, Technical Specification Group Radio Access Network. 3GPP (Release 18).

[Abdallah et al., 2024] Abdallah, A., Celik, A., Mansour, M. M., and Eltawil, A. M. (2024). Multi-agent deep reinforcement learning for beam codebook design in RIS-aided systems. *IEEE Transactions on Wireless Communications*, 23(7):7983–7999.

[Abdel-Aziz et al., 2020] Abdel-Aziz, M. K., Samarakoon, S., Liu, C.-F., Bennis, M., and Saad, W. (2020). Optimized age of information tail for ultra-reliable low-latency communications in vehicular networks. *IEEE Transactions on Communications*, 68(3):1911–1924.

[Alalewi et al., 2021] Alalewi, A., Dayoub, I., and Cherkaoui, S. (2021). On 5G-V2X use cases and enabling technologies: A comprehensive survey. *IEEE Access*, 9:107710–107737.

[Alkhateeb, 2019] Alkhateeb, A. (2019). DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications. *arXiv preprint arXiv:1902.06435*.

[Alkhateeb et al., 2023] Alkhateeb, A., Jiang, S., and Charan, G. (2023). Real-time digital twins: Vision and research directions for 6G and beyond. *IEEE Communications Magazine*, 61(11):128–134.

[Andoni et al., 2015] Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., and

Schmidt, L. (2015). Practical and optimal LSH for angular distance. *Adv. Neural Inf. Process. Syst.*, 28.

[Atallah et al., 2016] Atallah, R., Khabbaz, M., and Assi, C. (2016). Energy harvesting in vehicular networks: a contemporary survey. *IEEE Wireless Communications*, 23(2):70–77.

[Bagheri et al., 2021] Bagheri, H., Noor-A-Rahim, M., et al. (2021). 5G NR-V2X: Toward connected and cooperative autonomous driving. *IEEE Commun. Standards Mag.*, 5(1):48–54.

[Barnard et al., 2022] Barnard, P., Marchetti, N., and DaSilva, L. A. (2022). Robust network intrusion detection through explainable artificial intelligence (XAI). *IEEE Netw. Lett.*, 4(3):167–171.

[Ben Saad et al., 2022] Ben Saad, S., Brik, B., and Ksentini, A. (2022). A trust and explainable federated deep learning framework in zero touch B5G networks. In *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pages 1037–1042.

[Brik et al., 2023] Brik, B. et al. (2023). A survey on explainable ai for 6G O-RAN: Architecture, use cases, challenges and research directions. *arXiv preprint arXiv:2307.00319*.

[Celik and Eltawil, 2024] Celik, A. and Eltawil, A. M. (2024). At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence. *IEEE Open Journal of the Communications Society*, 5:2433–2489.

[Chandrashekar and Sahin, 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & electrical engineering*, 40(1):16–28.

[Choi et al., 2020] Choi, J., Choi, J., and Rhee, W. (2020). Interpreting neural ranking models using grad-cam.

[Cousik, 2022] Cousik, T. S. a. (2022). Deep learning for fast and reliable initial access in AI-driven 6G mmWave networks. *IEEE Transactions on Network Science and Engineering*.

[DeepMIMO,] DeepMIMO. DeepMIMO Dataset Generation. `https://deepmimo.net/`. Accessed: Nov. 25, 2024.

[Demirhan and Alkhateeb, 2022] Demirhan, U. and Alkhateeb, A. (2022). Radar aided 6g beam prediction: Deep learning algorithms and real-world demonstration. In *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, pages 2655–2660. IEEE.

[Ding et al., 2022] Ding, G., Yuan, J., Yu, G., and Jiang, Y. (2022). Two-timescale resource management for ultrareliable and low-latency vehicular communications. *IEEE Transactions on Communications*, 70(5):3282–3294.

[Dong et al., 2019] Dong, R., She, C., Hardjawana, W., Li, Y., and Vucetic, B. (2019). Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin. *IEEE Trans. Wireless Commun.*, 18(10):4692–4707.

[Dong et al., 2022] Dong, Z., Zhu, X., Jiang, Y., Zeng, H., Wei, Z., Zheng, F.-C., and Leung, K.-C. (2022). Dynamic manager selection assisted resource allocation in urllc with finite block length for 5G-V2X platoons. *IEEE Transactions on Vehicular Technology*, 71(11):11336–11350.

[Durisi et al., 2016] Durisi, G., Koch, T., and Popovski, P. (2016). Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proceedings of the IEEE*, 104(9):1711–1726.

[Fang et al., 2022] Fang, C.-H., Feng, K.-T., and Yang, L.-L. (2022). Resource allocation for urllc service in in-band full-duplex-based V2I networks. *IEEE Transactions on Communications*, 70(5):3266–3281.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1126–1135. PMLR.

[Fu et al., 2021] Fu, X., Guo, C., Qu, Y., and Lin, X.-H. (2021). Resource allocation and blocklength selection for low-latency vehicular communications. *IEEE Wireless Communications Letters*, 10(5):914–918.

[Garcia et al., 2021a] Garcia, M. H. C. et al. (2021a). A tutorial on 5G NR V2X communications. *IEEE Communications Surveys & Tutorials*, 23(3):1972–2026.

[Garcia et al., 2021b] Garcia, M. H. C., Molina-Galan, A., Boban, M., Gozalvez, J., Coll-Perales, B., Şahin, T., and Kousaridas, A. (2021b). A tutorial on 5G NR V2X communications. *IEEE Communications Surveys and Tutorials*, 23(3):1972–2026.

[Giordani et al., 2018] Giordani, M., Polese, M., Roy, A., Castor, D., and Zorzi, M. (2018). A tutorial on beam management for 3gpp nr at mmwave frequencies. *IEEE Communications Surveys and Tutorials*, 21(1):173–196.

[Goodfellow et al., 2014] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples.

[Gu et al., 2024] Gu, Y., She, C., Bi, S., Quan, Z., and Vucetic, B. (2024). Graph neural network for distributed beamforming and power control in massive urllc networks. *IEEE Transactions on Wireless Communications*, pages 1–1.

[Guo et al., 2023] Guo, C., Wang, C., Cui, L., Zhou, Q., and Li, J. (2023). Radio resource management for C-V2X: From a hybrid centralized-distributed scheme to a distributed scheme. *IEEE Journal on Selected Areas in Communications*, 41(4):1023–1034.

[Guo, 2020] Guo, W. (2020). Explainable artificial intelligence for 6G: Improving trust between human and machine. *IEEE Communications Magazine*, 58:39–45.

[Hamon et al., 2020] Hamon, R., Junklewitz, H., Sanchez, I., et al. (2020). Robustness and explainability of artificial intelligence. *Publications Office of the European Union*, 207:2020.

[Hazarika et al., 2023] Hazarika, B. et al. (2023). Radit: Resource allocation in digital twin-driven UAV-aided internet of vehicle networks. *IEEE Journal on Selected Areas in Communications*, 41(11):3369–3385.

[Heath et al., 2016] Heath, R. W., Gonzalez-Prelcic, N., Rangan, S., Roh, W., and Sayeed, A. M. (2016). An overview of signal processing techniques for millimeter wave MIMO systems. *IEEE Journal of Selected Topics in Signal Processing*, 10(3):436–453.

[Heng and Andrews, 2021] Heng, Y. and Andrews, J. G. (2021). Machine learning-assisted beam alignment for mmWave systems. *IEEE Transactions on Cognitive Communications and Networking*, 7(4):1142–1155.

[Heng et al., 2022] Heng, Y., Mo, J., and Andrews, J. G. (2022). Learning site-specific probing beams for fast mmWave beam alignment. *IEEE Transactions on Wireless Communications*, 21(8):5785–5800.

[Hu et al., 2019] Hu, Y., Zhu, Y., Gursoy, M. C., and Schmeink, A. (2019). Swipt-enabled relaying in IoT networks operating with finite blocklength codes. *IEEE Journal on Selected Areas in Communications*, 37(1):74–88.

[Hua et al., 2024] Hua, M. et al. (2024). Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects.

[Jalali et al., 2024] Jalali, J., Roa, J., Song, Y., Zhao, R., and Sheen, B. (2024). Fast best beam prediction and overhead reduction for 6G networks: A deep learning approach. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, pages 01–06. IEEE.

[Jiang and Alkhateeb, 2023] Jiang, S. and Alkhateeb, A. (2023). Digital twin based beam prediction: Can we train in the digital world and deploy in reality? In *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, pages 36–41. IEEE.

[Khan and Coleri, 2022] Khan, N. and Coleri, S. (2022). Resource allocation for ultra-reliable low-latency vehicular networks in finite blocklength regime. In *Proc. IEEE Int. Mediterranean Conf. Commun. Netw. (MeditCom)*, pages 322–327.

[Khan and Coleri, 2024] Khan, N. and Coleri, S. (2024). Event-triggered reinforcement learning based joint resource allocation for ultra-reliable low-latency V2X communications. *IEEE Transactions on Vehicular Technology*, pages 1–16.

[Khan et al., 2024] Khan, N., Coleri, S., Abdallah, A., Celik, A., and Eltawil, A. M. (2024). Explainable and robust artificial intelligence for trustworthy resource management in 6G networks. *IEEE Communications Magazine*, 62(4):50–56.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[Kumar et al., 2021] Kumar, A. S., Zhao, L., and Fernando, X. (2021). Multi-agent deep reinforcement learning-empowered channel allocation in vehicular networks. *IEEE Transactions on Vehicular Technology*, 71(2):1726–1736.

[Kyösti et al., 2007] Kyösti, P. et al. (2007). Ist-4-027756 winner II D1.1.2 V1.2 winner II channel models.

[Li et al., 2020] Li, C., Guo, W., et al. (2020). Trustworthy deep learning in 6G-enabled mass autonomy: From concept to quality-of-trust key performance indicators. *IEEE Vehicular Technology Magazine*, 15(4):112–121.

[Li et al., 2024] Li, J., Niu, Y., Wu, H., Ai, B., Quek, T. Q. S., Wang, N., and Chen, S. (2024). Effective joint scheduling and power allocation for urllc-oriented V2I communications. *IEEE Transactions on Vehicular Technology*, pages 1–12.

[Li et al., 2022] Li, X., Lu, L., Ni, W., Jamalipour, A., Zhang, D., and Du, H. (2022). Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications. *IEEE Transactions on Vehicular Technology*, 71(8):8810–8824.

[Li et al., 2023] Li, Y., Liu, F., Du, Z., Yuan, W., and Masouros, C. (2023). ISAC-enabled V2I networks based on 5G NR: How much can the overhead be reduced? In *2023 IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, pages 691–696. IEEE.

[Liang et al., 2019] Liang, L., Ye, H., and Li, G. Y. (2019). Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. *IEEE Journal on Selected Areas in Communications*, 37(10):2282–2292.

[Liang et al., 2017] Liang, Q., Que, F., and Modiano, E. (2017). Accelerated primal-dual policy optimization for safe reinforcement learning. In *Proc. Neural Inf. Process. Syst. (NIPS)*.

[Liu et al., 2024] Liu, P., Cui, H., and Zhang, N. (2024). DDQN-based centralized spectrum allocation and distributed power control for V2X communications. *IEEE Transactions on Vehicular Technology*, pages 1–12.

[Lu et al., 2023] Lu, J., Han, L., Wei, Q., Wang, X., Dai, X., and Wang, F.-Y. (2023). Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(4):2821–2831.

[Lundberg et al., 2020] Lundberg, S. M. et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, 2(1):56–67.

[Lundberg and Lee, 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst. 30*, pages 4765–4774.

[Luo and Alkhateeb, 2024] Luo, H. and Alkhateeb, A. (2024). Digital twin aided compressive sensing: Enabling site-specific MIMO hybrid precoding. *arXiv preprint arXiv:2405.07115*.

[Luong et al., 2019] Luong, N. C. et al. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys and Tutorials*, 21(4):3133–3174.

[Mnih et al., 2015] Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[Molina-Masegosa and Gozalvez, 2017] Molina-Masegosa, R. and Gozalvez, J. (2017). LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39.

[Morais and Alkhateeb, 2024] Morais, J. and Alkhateeb, A. (2024). Localization in digital twin MIMO networks: A case for massive fingerprinting. *arXiv preprint arXiv:2403.09614*.

[Nasir, 2020] Nasir, A. A. (2020). Min-Max decoding-error probability-based resource allocation for a URLLC system. *IEEE Communications Letters*, 24(12):2864–2867.

[Nasir and Guo, 2019] Nasir, Y. S. and Guo, D. (2019). Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. *IEEE Journal on Selected Areas in Communications*, 37(10):2239–2250.

[Niculescu-Mizil and Caruana, 2005] Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632.

[Pan et al., 2019] Pan, C., Ren, H., Deng, Y., Elkashlan, M., and Nallanathan, A. (2019). Joint blocklength and location optimization for URLLC-enabled uav relay systems. *IEEE Communications Letters*, 23(3):498–501.

[Papernot and McDaniel, 2018] Papernot, N. and McDaniel, P. (2018). Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

[Parvini et al., 2024] Parvini, M. et al. (2024). Resource allocation in V2X networks: From classical optimization to machine learning-based solutions. *IEEE Open Journal of the Communications Society*, 5:1958–1974.

[Polyanskiy et al., 2010] Polyanskiy, Y., Poor, H. V., and Verdu, S. (2010). Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359.

[Popovski et al., 2019] Popovski, P. et al. (2019). Wireless access in ultra-reliable low-latency communication (URLLC). *IEEE Transactions on Communications*, 67(8):5783–5801.

[Qi et al., 2020] Qi, C., Chen, K., Dobre, O. A., and Li, G. Y. (2020). Hierarchical codebook-based multiuser beam training for millimeter wave massive mimo. *IEEE Transactions on Wireless Communications*, 19(12):8142–8152.

[Rashid et al., 2020] Rashid, T. et al. (2020). Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.*, 21(178):1–51.

[Rawal et al., 2022] Rawal, A. et al. (2022). Recent advances in trustworthy explainable artificial intelligence: Status, challenges, and perspectives. *IEEE Transactions on Artificial Intelligence*, 3(6):852–866.

[Remcom,] Remcom. Remcom Wireless InSite. `http://www.remcom.com/wireless-insite`. Accessed: Nov. 25, 2024.

[Ren et al., 2020a] Ren, H., Pan, C., Deng, Y., Elkashlan, M., and Nallanathan, A. (2020a). Joint power and blocklength optimization for URLLC in a factory automation scenario. *IEEE Transactions on Wireless Communications*, 19(3):1786–1801.

[Ren et al., 2020b] Ren, H., Pan, C., Deng, Y., Elkashlan, M., and Nallanathan, A. (2020b). Resource allocation for secure urllc in mission-critical IoT scenarios. *IEEE Transactions on Communications*, 68(9):5793–5807.

[Rezaie et al., 2021] Rezaie, S., Amiri, A., de Carvalho, E., and Manchón, C. N. (2021). Deep transfer learning for location-aware millimeter wave beam selection. *IEEE Communications Letters*, 25(9):2963–2967.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier.

[Sabih et al., 2020] Sabih, M. et al. (2020). Utilizing explainable AI for quantization and pruning of deep neural networks. *arXiv:2008.09072*.

[She et al., 2018] She, C. et al. (2018). Improving network availability of ultra-reliable and low-latency communications with multi-connectivity. *IEEE Transactions on Communications*, 66(11):5482–5496.

[She et al., 2021] She, C., Sun, C., Gu, Z., Li, Y., Yang, C., Poor, H. V., and Vucetic, B. (2021). A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning. *Proc. IEEE*, 109(3):204–246.

[Shirvanimoghaddam et al., 2019] Shirvanimoghaddam, M., Mohammadi, M. S., Abbas, R., Minja, A., Yue, C., Matuz, B., Han, G., Lin, Z., Liu, W., Li, Y., Johnson, S., and Vucetic, B. (2019). Short block-length codes for ultra-reliable low latency communications. *IEEE Communications Magazine*, 57(2):130–137.

[Shrikumar et al., 2017] Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 3145–3153. PMlR.

[Solowjow and Trimpe, 2020] Solowjow, F. and Trimpe, S. (2020). Event-triggered learning. *Automatica*, 117:109009.

[Sun et al., 2016] Sun, W., Strom, E. G., Brannstrom, F., Sou, K. C., and Sui, Y. (2016). Radio resource management for D2D-based V2V communication. *IEEE Transactions on Vehicular Technology*, 65(8):6636–6650.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Tan et al., 2021] Tan, J., Liang, Y.-C., Zhang, L., and Feng, G. (2021). Deep reinforcement learning for joint channel selection and power control in D2D networks. *IEEE Transactions on Wireless Communications*, 20(2):1363–1378.

[Terra et al., 2020] Terra, A., Inam, R., et al. (2020). Explainability methods for identifying root-cause of SLA violation prediction in 5G network. In *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pages 1–7. IEEE.

[Tian et al., 2022] Tian, J. et al. (2022). Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks. *IEEE Internet Things J.*, 9(3):1683–1695.

[Tseng, 2001] Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109:475–494.

[Vinitsky et al., 2018] Vinitsky, E., Kreidieh, A., Le Flem, L., Kheterpal, N., Jang, K., Wu, C., Wu, F., Liaw, R., Liang, E., and Bayen, A. M. (2018). Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Proc. Conf. Robot Learn.*, pages 399–409. PMLR.

[Wang et al., 2020a] Wang, K. et al. (2020a). Packet error probability and effective throughput for ultra-reliable and low-latency UAV communications. *IEEE Transactions on Communications*, 69(1):73–84.

[Wang et al., 2020b] Wang, X., Zhang, Y., Shen, R., Xu, Y., and Zheng, F.-C. (2020b). Drl-based energy-efficient resource allocation frameworks for uplink noma systems. *IEEE Internet of Things J.*, 7(8):7279–7294.

[Wu et al., 2021] Wu, W., Liu, R., Yang, Q., Shan, H., and Quek, T. Q. S. (2021). Learning-based robust resource allocation for ultra-reliable V2X communications. *IEEE Transactions on Wireless Communications*, 20(8):5199–5211.

[Xiang et al., 2021] Xiang, P. et al. (2021). Multi-agent RL enables decentralized spectrum access in vehicular networks. *IEEE Transactions on Vehicular Technology*, 70(10):10750–10762.

[Xue et al., 2024a] Xue, Q. et al. (2024a). AI/ML for beam management in 5G-advanced: A standardization perspective. *IEEE Vehicular Technology Magazine*, pages 2–10.

[Xue et al., 2024b] Xue, Q., Ji, C., Ma, S., Guo, J., Xu, Y., Chen, Q., and Zhang, W. (2024b). A survey of beam management for mmwave and thz communications towards 6G. *IEEE Communications Surveys and Tutorials*, 26(3):1520–1559.

[Yang et al., 2020a] Yang, H., Zhang, K., Zheng, K., and Qian, Y. (2020a). Joint frame design and resource allocation for ultra-reliable and low-latency vehicular networks. *IEEE Transactions on Wireless Communications*, 19(5):3607–3622.

[Yang et al., 2020b] Yang, H., Zheng, K., Zhang, K., Mei, J., and Qian, Y. (2020b). Ultra-reliable and low-latency communications for connected vehicles: Challenges and solutions. *IEEE Network*, 34(3):92–100.

[Yang et al., 2019] Yang, X., He, H., and Liu, D. (2019). Event-triggered optimal neuro-controller design with reinforcement learning for unknown nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(9):1866–1878.

[Ye et al., 2019] Ye, H., Li, G. Y., and Juang, B.-H. F. (2019). Deep reinforcement learning based resource allocation for V2V communications. *IEEE Transactions on Vehicular Technology*, 68(4):3163–3173.

[Yuan et al., 2021] Yuan, Y., Zheng, G., Wong, K.-K., and Letaief, K. B. (2021). Meta-reinforcement learning based resource allocation for dynamic V2X communications. *IEEE Transactions on Vehicular Technology*, 70(9):8964–8977.

[Zappone et al., 2019] Zappone, A., Renzo, M. D., Debbah, M., Lam, T. T., and Qian, X. (2019). Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization. *IEEE Vehicular Technology Magazine*, 14(3):60–69.

[Zelenbaba et al., 2022] Zelenbaba, S., Rainer, B., Hofer, M., and Zemen, T. (2022). Wireless digital twin for assessing the reliability of vehicular communication links. In *Proc. IEEE Global Commun. Conf. Workshops (GC Wkshps)*, pages 1034–1039. IEEE.

[Zhang et al., 2022] Zhang, T. et al. (2022). Interpreting AI for networking: Where we are and where we are going. *IEEE Communications Magazine*, 60(2):25–31.

[Zhang et al., 2024] Zhang, T., Wang, Y., Ouyang, M., Xing, L., and Gao, F. (2024). Third-party camera aided beam alignment real-world prototype for mmwave communications. In *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, pages 1–6. IEEE.

[Zhang et al., 2020] Zhang, X., Peng, M., Yan, S., and Sun, Y. (2020). Deep-

reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. *IEEE Internet of Things Journal*, 7(7):6380–6391.

[Zheng et al., 2022] Zheng, Y., Lin, L., Zhang, T., Chen, H., Duan, Q., Xu, Y., and Wang, X. (2022). Enabling robust DRL-driven networking systems via teacher-student learning. *IEEE J. Sel. Areas Commun.*, 40(1):376–392.

[Zhu et al., 2022] Zhu, Y., Hu, Y., Yuan, X., Cenk Gursoy, M., Vincent Poor, H., and Schmeink, A. (2022). Joint convexity of error probability in blocklength and transmit power in the finite blocklength regime. *IEEE Transactions on Wireless Communications*, pages 1–1.

[Zhuang et al., 2021] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2021). A comprehensive survey on transfer learning. *Proc. IEEE.*, 109(1):43–76.