PhD-FSTM-2025-125

Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 28 October 2025 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

by

Ke HE
Born on 2 June 1993 in Guangxi, China

RISK-AWARE INTELLIGENCE FOR LARGE-SCALE PARTIALLY OBSERVABLE COMMUNICATION SYSTEMS

Dissertation defence committee
Dr. Symeon CHATZINOTAS, dissertation supervisor
Professor, Université du Luxembourg

Dr. Bhavani Shankar MYSORE RAMA RAO, Chairman
Assistant Professor, Université du Luxembourg

Dr. Beatriz SORET, Member
Associate Professor, Universidad de Málaga, Spain

Dr. Thang Xuan VU, Member
Research scientist, Université du Luxembourg

Dr. Viet PHAM, Member
Assistant Professor, Trinity College Dublin, Ireland

Affidavit / Statement of originality

I declare that this thesis:

- is the result of my own work. Any contribution from any other party, and any use of generative artificial intelligence technologies have been duly cited and acknowledged;
- is not substantially the same as any other that I have submitted, and;
- is not being concurrently submitted for a degree, diploma or other qualification at the University of Luxembourg or any other University or similar institution except as specified in the text.

With my approval I furthermore confirm the following:

- I have adhered to the rules set out in the University of Luxembourg's Code of Conduct and the Doctoral Education Agreement (DEA)[1], in particular with regard to Research Integrity.
- I have documented all methods, data, and processes truthfully and fully.
- I have mentioned all the significant contributors to the work.
- I am aware that the work may be screened electronically for originality.

I acknowledge that if any issues are raised regarding good research practices based on the review of the thesis, the examination may be postponed pending the outcome of any investigation of such issues. If a degree was conferred, any such subsequently discovered issues may result in the cancellation of the degree.

Approved on 2025-09-29

[1] If applicable (DEA is compulsory since August 2020)

Abstract

The evolution towards Sixth-Generation (6G) communication systems is characterized by an unprecedented increase in scale, decentralization, and dynamism. This trend is evident by terrestrial networks employing massive Multiple-Input Multiple-Output (MIMO) antenna arrays and non-terrestrial networks composed of ultra-dense Low Earth Orbit (LEO) satellite constellations. In such large-scale environments, obtaining a complete and timely view of the true system state, e.g., complete channel state information (CSI) or global network state, is often infeasible due to prohibitive communication overhead and physical constraints. This gives rise to a fundamental challenge we term *partial observability at scale*.

Traditional control and optimization methods, which typically assume complete system knowledge, fail to provide robust solutions in such environments. Crucially, these methods fail because they do not adequately handle the risks of partial observability. Many frameworks are *risk-oblivious*, optimizing for average performance while ignoring critical QoS degradation caused by incomplete state knowledge. Even recent constrained approaches are *risk-myopic*, focusing on average performance constraints while failing to control for high-impact tail-end events like severe latency spikes or QoS breaches. This leaves the system vulnerable to unacceptable performance violations.

Recognizing this gap, this dissertation argues that there is an urgent need for intelligent and autonomous decision-making frameworks capable of operating reliably under partial observability at scale and managing the associated risks effectively. This leads to the central research problem addressed herein: *How can communication agents make robust and risk-aware decisions in large-scale partially observable communication systems?* To answer this question, this dissertation moves beyond risk-oblivious and risk-myopic approaches by proposing a unified framework for *risk-aware intelligence in large-scale partially observable communication systems*. We develop this framework through two complementary paradigms: model-based risk-aware planning and model-free risk-aware reinforcement learning, demonstrated on antenna selection in massive MIMO with partial CSI and asynchronous packet routing in LEO mega-constellations, respectively.

Part I of the dissertation introduces the model-based risk-aware planning paradigm, instantiated for antenna selection in massive MIMO systems with incomplete CSI. The core of this approach is a novel *Risk-Aware Monte-Carlo Tree Search (RA-MCTS)* planner. RA-MCTS leverages a predictive *"world model"* that learns to forecast the full CSI from partial historical measurements, planning over the resulting belief distribution of future channel states to select actions that explicitly minimize the risk of QoS violations. To enhance the predictive accuracy of this framework, we subsequently develop an advanced spatio-temporal world model featuring a novel *Crossover Attention (XOA)* mechanism, which modifies the standard Transformer architecture to explicitly capture both spatial and temporal correlations. Validated in the massive MIMO context, this model-based framework demonstrates significant power reduction while drastically lowering the risk of QoS breaches compared to state-of-the-art baselines.

Part II of the dissertation presents the model-free risk-aware multi-agent reinforcement learning (MARL) paradigm, designed for decentralized complex systems where learning an accurate world model is intractable. We demonstrate this in the challenging domain of asynchronous packet routing in LEO satellite mega-constellations. To address this, we introduce *Principled Risk-aware Independent Multi-Agent Learning (PRIMAL)*, a framework designed for asynchronous environments that enables each satellite to make independent event-driven routing decisions. To manage the risks arise from uncoordinated actions under partially observability, PRIMAL employs a principled distributional primal-dual learning method. By learning the full conditional distribution of routing outcomes (e.g., delay), agents directly constrain the *Conditional-Value-at-Risk (CVaR)*, enabling robust control over worst-case risks like severe latency spikes and network congestion. When applied to routing in a high-fidelity LEO constellation simulation, PRIMAL successfully mitigates these issues, significantly outperforming baselines.

By developing both model-based and model-free paradigms, this dissertation provides a comprehensive set of tools to address the challenge of partial observability at scale. The key contribution lies in moving beyond heuristic or average-case optimization to provide principled methods for managing worst-case performance risks. The developed algorithms are not merely solutions to specific problems but are concrete demonstrations of a foundational approach for engineering the robust, efficient, and truly intelligent communication systems of the next generation.

Acknowledgments

First and foremost, I would like to express my deepest and most sincere gratitude to my supervisor, Prof. Symeon Chatzinotas and Dr. Thang Xuan Vu. Throughout this Ph.D. journey, your invaluable guidance, consistent encouragement, and sharp academic insight have been the most important support for my research. Your mentorship not only shaped the direction and depth of my work but also significantly contributed to my growth as an independent and critical researcher. I am truly grateful for the freedom and trust you gave me to explore my own ideas, even when they diverged from conventional paths. Your belief in my potential has been a constant source of motivation.

I would also like to extend my heartfelt thanks to my thesis defense committee, including Prof. Beatriz Soret, Prof. Viet Pham, Prof. Bhavani Shankar, Dr. Thang Xuan Vu and Prof. Symeon Chatzinotas, for their time, expertise, and thought-provoking questions. Their rigorous examination and valuable feedback have been a meaningful and rewarding final step in this academic endeavor, helping me see my work in new lights and preparing me for future challenges.

I would also like to sincerely thank Prof. Björn Ottersten (University of Luxembourg), Prof. George K. Karagiannidis (Aristotle University of Thessaloniki), Prof. Dinh Thai Hoang (University of Technology Sydney), Prof. Xianfu Lei (Southwest Jiaotong University), and Prof. Lisheng Fan (Guangzhou University) for their valuable suggestions, constructive comments, and generous guidance on various aspects of my research. Their support has greatly enriched the depth and quality of this dissertation.

I am equally grateful to my colleagues and friends at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. The free, lovely and collaborative research environment, the stimulating discussions, and the shared moments of struggle and success have made my Ph.D. experience truly memorable. I deeply appreciate the engaging discussions, technical exchanges, and countless moments that we shared, whether in the lab, at academic events, or over casual coffee breaks. These interactions have not only sharpened my thinking but also brought joy to the everyday process of research.

Special thanks go to the administrative and technical staff at SnT and the

University of Luxembourg, whose efficiency and dedication ensured that all the behind-the-scenes processes, whether related to logistics, travel, or computing infrastructure, all ran smoothly throughout my doctoral journey.

Last but certainly not least, I owe my deepest gratitude to my family. Their unconditional love and belief in me, and endless support have been the strongest shoulder of my academic journey. Through the inevitable highs and lows, their encouragement and patience have carried me forward. To my parents and brother, who have always supported my dreams even from afar. Without your sacrifices and faith, this dissertation would not have been carried out.

This milestone is not mine alone. It is the culmination of the contributions, support, and goodwill of many people. Personally, it marks not just the conclusion of one journey, but the beginning of a new chapter, the one that I embrace with the quote:

路漫漫其修远兮　吾将上下而求索

Long long had been my road and far far was the journey
I would go up and down to seek my soul desire

何 科
HE Ke
July 2025, Luxembourg

Perface

This Ph.D. dissertation has been carried out from December, 2021 to July, 2025 at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), Université du Luxembourg, Luxembourg, under the supervision of Prof. Symeon Chatzinotas and Dr. Thang Xuan Vu at SnT, Université du Luxembourg, Luxembourg. The time-to-time evaluation of the Ph.D. dissertation was duly performed by the CET members constituting the supervisors at SnT, Université du Luxembourg, Luxembourg and Prof. Beatriz Soret from Universidad de Málaga.

Support of the Dissertation

This Ph.D. Thesis has been supported in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references FNR/C19/IS/13718904/ASWELL, FNR/C22/IS/17220888/RUTINE, INTER/23/17941203/PASSIONATE, and INTER/-MOBILITY/2023/IS/18014377/MCR. Additionally, the time-to-time support from SIGCOM is also gratefully acknowledged.

Contents

1 Introduction

1.1 Motivation

The rapid evolution of wireless communication networks, particularly in the transition toward Sixth-Generation (6G) systems, is marked by unprecedented scale, decentralization, and heterogeneity [1, 2, 3, 4]. The next-generation networks are visioned to support ultra-dense connectivity, strict latency and reliability requirements, and dynamic multi-domain coordination across terrestrial and non-terrestrial infrastructures [5, 6, 7]. The defining characteristic of next-generation networks is their massive scale[8, 9, 10]. In the terrestrial domain, base stations are being equipped with increasingly large antenna arrays to implement massive multiple-input multiple-output (MIMO) communication[11, 12, 9], while in the non-terrestrial domain, large fleets of low Earth Orbit (LEO) satellites are being deployed to form low-latency global networks[3, 13, 14]. These trends are transforming communication networks from static centrally coordinated systems into highly dynamic distributed infrastructures.

In such large-scale settings, intelligent decision-making becomes a fundamental requirement[6, 15, 16, 17, 18]. Centralized control is no longer viable due to communication overhead and latency limitations. Instead, network entities must operate autonomously and adaptively, making local decisions based on partial and noisy information[19, 20]. The scale and dynamics of the system introduce challenges in both observability and coordination, with requirements for performance,

reliability, and robustness[21]. Moreover, the operational environments are often stochastic and resource-constrained. Channel conditions fluctuate rapidly, user demands vary unpredictably, and link failures or congestion may occur without warning[22, 23, 24]. For example, decision making processes such as antenna selection at a base station or packet routing at a satellite must be made under incomplete knowledge of the system state that may lead to unexpected serious outcomes[25, 26, 27, 28, 29]. Addressing these challenges requires a new class of decision-making frameworks that are not only adaptive and decentralized but also explicitly *risk-aware*.

This dissertation aims to study and response to this emergent need, with a specific focus on two representative decision-making problems in large-scale partially observable communication systems: *antenna selection in massive Multiple-Input Multiple-Output (MIMO) systems* (Chapters 3-4) and *packet routing in ultra-dense Low Earth Orbit (LEO) constellations* (Chapter 5). The two concrete cases precisely reflect the core challenges and diversity of next-generation communication infrastructures, and can serve as testbeds for developing and evaluating general-purpose risk-aware learning methods for robust communications and networking.

1.2 The Challenge: Partial Observability at Scale

1.2.1 Existing Works & Limitations

Existing research on antenna selection and LEO packet routing has made significant progress, yet it often builds on simplifying assumptions that are increasingly challenged by the scale and dynamics of next-generation networks. These assumptions typically relate to the availability of information (observability) and the nature of system operations (decentralization), leading to frameworks that are either risk-oblivious or risk-myopic in the presence of partial observations.

In the case of antenna selection, traditional approaches have primarily relied

on the availability of complete and instantaneous Channel State Information (CSI) to optimize metrics such as throughput, Signal-to-Noise-Ratio (SNR), or energy efficiency [30]. However, in large-scale multiuser MIMO systems, the acquisition of full CSI is a fundamental bottleneck. The associated pilot overhead of full CSI becomes prohibitive, consuming valuable resources and reducing the effective data transmission rate [31, 32]. Recognizing this, recent works have begun to address antenna selection with incomplete CSI. Some have formulated the problem as a Partially Observable Markov Decision Process (POMDP) [33, 34], while others have employed online learning techniques such as multi-armed bandits [31]. While these represent important steps forward, they are *risk-oblivious* and often fall short in practical scenarios. This is because they all fail to explicitly model the risk of violating long-term performance constraints, which can lead to intermittent QoS degradation despite good average performance. [32].

For LEO satellite packet routing, a similar pattern emerges. Early designs often assumed predictable network conditions, leveraging static topology snapshots or centralized contact planning to pre-calculate routes [22, 35]. Such methods are inherently frangible and *risk-oblivious*, as they cannot adapt to unpredictable real-time dynamics like traffic congestion, a primary cause of performance degradation. Data-driven Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful paradigm to address this dynamism by enabling decentralized decision-making [29]. However, in these frameworks, each satellite agent makes routing decisions based only on its local observations, such as the state of its own queue and immediate neighbors. This limited viewpoint creates a critical challenge: an action that appears optimal locally (e.g., forwarding a packet to the neighbor on the shortest path to the destination) can inadvertently contribute to downstream congestion, degrading global network performance. Existing Constrained Reinforcement Learning (CRL) frameworks attempt to manage this by optimizing objectives subject to constraints. While this introduces a certain level of *risk-awareness*, they tend to be *risk-myopic* [36]. They typically focus

on constraining *average* performance metrics (e.g., average delay) while failing to control for high-impact, tail-end events such as severe latency spikes, which could lead to unmanaged congestion in a decentralized system [37].

1.2.2 Research Problem

From a decision-making perspective, the fundamental challenge in both settings is the *partial observability at scale*. In massive MIMO, only partial or delayed CSI is available, leading to uncertain antenna utility [32]. In LEO networks, each satellite observes only its local queue and the link states of its direct neighbors, while the global network load and future connectivity remain uncertain [37]. These conditions create discrepancies between the true system state and the agent's perception, and thus require decision-making based on estimated or inferred global views, i.e., *an explicit or implicit belief over the true system state*. Hence, risk in this dissertation refers to the probability or severity of constraint violations due to imperfect inference of the system state. Mitigating such risk requires explicitly modeling the uncertainty arising from partial observations and accounting for its impact in learning and planning. This gives rise to a central research problem:

Research Problem

How can communication agents make robust risk-aware decisions under large-scale partial observability?

Specifically, how can we design principled methods that explicitly model the relationship between partial observations, belief estimation, uncertainty, and risk in order to make constraint-satisfying high-performance decisions?

1.3 Objectives & Methodologies

The primary objective of this dissertation is to answer the central research question by developing a unified and principled framework for *risk-aware intelligence*

in large-scale partially observable communication systems. We aim to design and validate algorithms that enable autonomous agents to learn robust sequential decision-making policies that optimize long-term performance while satisfying operational constraints under uncertainty. To achieve this, we investigate two complementary methodologies, each tailored to different assumptions about the system and available data:

- **Model-Based Risk-Aware Planning:** This methodology, forming Part I of the dissertation, is designed for scenarios where the underlying system dynamics can be effectively learned and simulated. It involves constructing a *world model* to forecast system evolution and maintain a belief over hidden states. This belief is then utilized by a risk-aware planner, such as a constrained Monte-Carlo Tree Search (MCTS), to select actions that minimize the probability of future constraint violations. This approach is primarily instantiated through our work on antenna selection under incomplete CSI [32, 38].

- **Model-Free Risk-Aware Reinforcement Learning (RL):** This methodology, forming Part II, is designed for complex (multi-agent) environments where an accurate world model is difficult to learn. In this case, agents must learn robust policies directly from interaction with the black-boxed environment. To manage risk in a principled manner, we employ a CRL framework based on principled primal-dual learning. Critically, we extend this framework with *distributional reinforcement learning*, which moves beyond learning simple averages by capturing the entire probability distribution of the targeted Quality of Service (QoS) metric. This capability enables the direct optimization of worst-case performance and the mitigation of tail-end risks. This approach is demonstrated in the context of asynchronous multi-agent packet routing in LEO networks [37].

1.4 Summary of Contributions

The main contributions of this dissertation are as follows:

- We establish a principled constrained decision-making paradigm for large-scale partially observable communication systems by formally connecting an agent's *belief* over the hidden system state to the *risk* of violating performance constraints. By quantifying risk as a direct function of state uncertainty, this approach provides a principled foundation to move beyond existing *risk-oblivious* and *risk-myopic* methods and enables the design of verifiably robust risk-aware intelligent agents.

- We develop a novel model-based algorithm for massive MIMO antenna selection that combines a spatio-temporal channel predictor with a **Risk-Aware Monte-Carlo Tree Search (RA-MCTS)** planner. This approach effectively manages QoS violation risks by planning over a belief of the incomplete CSI [32, 38].

- To enhance the predictive accuracy of the model-based planner, we develop an advanced spatio-temporal "world model" featuring a novel *Crossover Attention (XOA)* mechanism. This mechanism enhances the standard Transformer architecture to explicitly and simultaneously capture both spatial and temporal correlations, enabling the planning agent to make more informed and robust decisions under partial observability [38].

- We propose a model-free decentralized algorithm for LEO satellite routing, named **Principled Risk-aware Independent Multi-Agent Learning (PRIMAL)**, which is uniquely designed for asynchronous event-driven environments like packet routing in LEO constellations. It employs a *distributional primal-dual learning* approach to explicitly constrain the *Conditional-Value-at-Risk (*CVaR*)* of the targeted QoS metric, thereby mitigating tail-end QoS constraint violation risk [37].

1.5 List of Publications

This dissertation is based on the following papers that have been published or are currently under review:

Journals

[J3] ***K. He***, L. He, L. Fan, X. Lei, T. X. Vu, G. K. Karagiannidis and S. Chatzinotas, "SCA-LLM: Spectral-Attentive Channel Prediction with Large Language Models in MIMO-OFDM," in *IEEE Journal of Selected Areas in Communications*, 2025, under review.

[J2] ***K. He***, T. X. Vu, L. Fan, S. Chatzinotas and B. Ottersten, "Spatio-Temporal Predictive Learning Using Crossover Attention for Communications and Networking Applications," in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 479-490, 2025.

[J1] ***K. He***, T. X. Vu, D. T. Hoang, D. N. Nguyen, S. Chatzinotas and B. Ottersten, "Risk-Aware Antenna Selection for Multiuser Massive MIMO Under Incomplete CSI," in *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 11001-11014, Sept. 2024.

Conferences

[C4] ***K. He***, T. X. Vu, L. He, L. Fan, S. Chatzinotas and B. Ottersten, "Asynchronous Risk-Aware Multi-Agent Routing for Ultra-Dense LEO Satellite Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2026, under review.

[C3] ***K. He***, T. X. Vu, S. Chatzinotas and B. Ottersten, "Spatio-Temporal Traffic Prediction Using Crossover Attention for Communications and Networking," in *IEEE Global Communications Conference Workshops (GLOBECOM Wkshps)*, 2024.

[C2] ***K. He***, T. X. Vu, S. Chatzinotas and B. Ottersten, "Fast Optimal Antenna Selection for Massive MIMO," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 186–190.

[C1] ***K. He***, T. X. Vu, S. Chatzinotas and B. Ottersten, "Learning-Based Joint Channel Prediction and Antenna Selection for Massive MIMO with Partial CSI," in *IEEE Global Communications Conference Workshops (GLOBECOM Wkshps)*, 2022, pp. 178–183.

1.6 Outline of the Dissertation

The remainder of this dissertation is organized as follows. **Chapter 2** introduces the necessary technical background on massive MIMO, LEO networks, reinforcement learning, and risk-sensitive decision-making.

Part I of the dissertation develops our model-based methodology. **Chapter 3** introduces our risk-aware planning framework for antenna selection under incomplete CSI. **Chapter 4** then explores the advanced spatio-temporal predictive models that aims to build a better world model for this framework.

Part II transitions to our model-free approach. **Chapter 5** presents the asynchronous MARL framework for LEO routing with QoS constraints on average costs. We then further extends this framework to mitigate tail-end risks by using distributional learning to constrain CVaR with the given risk-level.

Finally, **Chapter 6** summarizes the key findings of this dissertation and outlines promising directions for future research.

2 Preliminaries

This chapter provides the technical background necessary to understand the core concepts of this dissertation. We begin by formalizing the sequential decision-making problem using the framework of Markov Decision Process (MDP). We then review advanced planning and learning techniques, including MCTS and Deep Reinforcement Learning (DRL), which form the basis for our proposed risk-aware frameworks. Specifically, we will detail the mechanics of key DRL algorithms including Q-Learning and Soft Actor-Critic (SAC), and discuss their extension to multi-agent settings, as these will be referred to in later chapters as either performance baselines or core components of our developed solutions.

2.1 Sequential Decision-Making

The core of the problems addressed in this dissertation lies in *sequential decision-making under uncertainty and constraints*. To provide necessary background before we dive into the technical details, it is important to introduce the fundamental concepts that are common in sequential decision-making problems. Specifically, sequential decision-making refers that an agent interacts with an environment over a sequence of time steps, where its actions influence future states and the rewards it receives. This interaction loop is illustrated in Figure 2.1, and the Markov Decision Process (MDP) provides a formal mathematical framework for modeling such problems [17].

Formally, an episodic MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where:

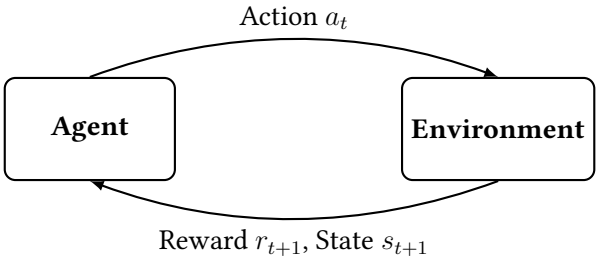

Figure 2.1 The agent-environment interaction loop in an MDP.

- $\mathcal{S}$ is the set of all possible states the environment can be in.
- $\mathcal{A}$ is the set of all possible actions the agent can take.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, where $\mathcal{P}(s'|s, a)$ gives the probability of transitioning to state s' after taking action a in state s.
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, where $r(s, a)$ is the immediate reward received after taking action a in state s.
- $\gamma \in [0, 1)$ is the discount factor, which determines the present value of future rewards. A value close to 0 prioritizes immediate rewards, while a value close to 1 gives more weight to long-term gains.

The agent's behavior is defined by a *policy*, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, where $\pi(a|s)$ is the probability of taking action a in state s. The goal of the agent is to learn an optimal policy, π^*, that maximizes the *return*, which is the expected sum of discounted future rewards. Starting from timestep t, the return is defined as $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$.

To evaluate policies, we use *value functions*. The *state-value function*, $V^\pi(s)$, is the expected return starting from state s and following policy π:

$$V^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s \right]. \tag{2.1}$$

The *action-value function*, $Q^\pi(s, a)$, is the expected return after taking action a in state s and then following policy π:

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s, A_t = a \right]. \quad (2.2)$$

These value functions satisfy recursive relationships known as the *Bellman equations*. The Bellman expectation equation for Q^π:

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}, a' \sim \pi}[Q^\pi(s', a')]. \quad (2.3)$$

The optimal policy π^* has a corresponding optimal action-value function $Q^*(s, a) = \max_\pi Q^\pi(s, a)$, which satisfies the *Bellman optimality equation* [39]:

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}}[\max_{a'} Q^*(s', a')]. \quad (2.4)$$

Solving an MDP means finding this optimal policy π^*. Once Q^* is known, the optimal policy is to greedily select the action with the highest Q-value in any given state.

2.2 Monte Carlo Tree Search

Monte-Carlo Tree Search (MCTS) is a powerful model-based heuristic search algorithm for sequential decision-making problems, particularly those with vast state spaces where exhaustive search is infeasible [40]. It intelligently explores the search space by constructing an asymmetric tree, where nodes represent states and edges represent actions. By iteratively simulating outcomes (rollouts) from the current state, MCTS effectively balances the exploration of new, unproven actions with the exploitation of actions already known to yield high rewards. This allows it to converge on near-optimal decisions without evaluating every possibility, making it particularly well-suited for combinatorial problems such

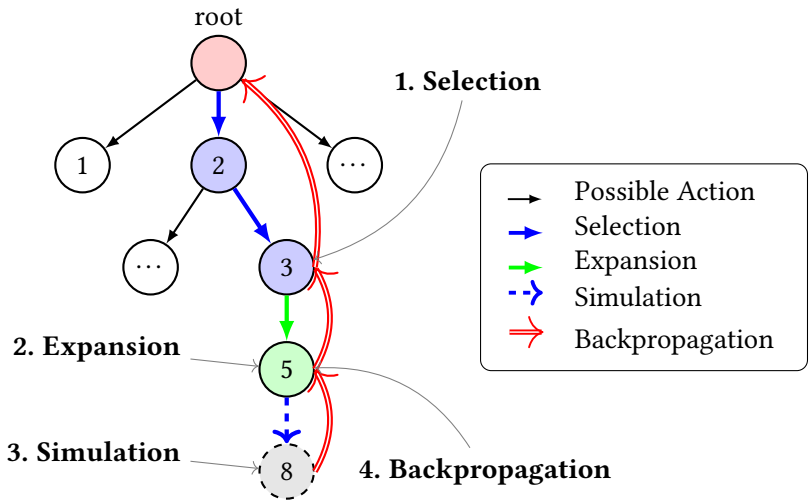

Figure 2.2 The four steps of the MCTS algorithm shown on a single tree: A path is **selected** using Upper Confidence bounds for Trees (UCT), a new node is **expanded**, a fast and often random **simulation** (rollout) is run, and the result is **backpropagated** to update node statistics.

as antenna selection in massive MIMO [25]. As illustrated in Figure 2.2, the algorithm cycles through the following four main steps:

1. **Selection:** The process begins at the root node (the current state) and recursively descends the tree. At each node, a tree policy is used to select the most promising child (action). A widely used and effective policy is the Upper Confidence bounds for Trees (UCT), which balances two key objectives. It selects a child node j that maximizes the following value:

$$\mathrm{UCT}_j = \underbrace{Q(s_j)}_{\text{Exploitation}} + \underbrace{C\sqrt{\frac{\ln N(s_p)}{N(s_j)}}}_{\text{Exploration}}, \tag{2.5}$$

   where $Q(s_j)$ is the child node's current estimated value (i.e., its average reward from previous simulations), encouraging the algorithm to *exploit* known good paths. The second term promotes *exploration*, where $N(s_p)$ is the visit count of the parent node, $N(s_j)$ is the visit count of the child node,

and $C > 0$ is a tunable exploration constant that weights the preference for less-visited nodes. This selection process continues until a leaf node (that has unvisited actions) is reached.

2. **Expansion:** When a leaf node is reached, the search tree is expanded. A new child node is created, corresponding to one of the previously unexplored actions from the current state. This new node is then added to the tree, and its statistics (e.g., visit count and value) are initialized.

3. **Simulation (Rollout):** From this newly expanded node, a simulation or rollout is conducted. This involves running a default policy until a terminal state is reached. The default policy is often a fast sub-optimal policy that selects actions randomly for computational efficiency. This rollout generates a complete trajectory from the new node, and the cumulative reward obtained provides a Monte Carlo estimate of the new node's value. This step is what makes MCTS a sampling-based method.

4. **Backpropagation:** The outcome of the simulation (i.e., the terminal reward) is then propagated back up the tree from the newly expanded node to the root. For every node along the selected path, the visit count $N(s)$ is incremented, and its value estimate $Q(s)$ is updated to incorporate the new simulation result.

By repeating these four steps for a specified number of iterations, i.e., a computational budget is reached, MCTS incrementally builds a search tree that is asymmetrically focused on the most promising regions of the state-action space. The value estimates for nodes closer to the root become progressively more accurate, allowing the agent to make a well-informed final decision by selecting the best action from the root that leads to the most-visited or highest-valued child node. The general procedure is detailed in Algorithm 1. Clearly, its ability to handle uncertainty through repeated sampling makes it an ideal backbone for the model-based risk-aware planning framework developed in this dissertation.

Algorithm 1: Monte Carlo Tree Search (MCTS)

```
1  function MCTS(state)
2      Create root node v_0 with state s_0 = state;
3      for i = 1 to num_iterations do
4          v_l ← TreePolicy(v_0) ;          // Selection and Expansion
5          Δ ← DefaultPolicy(s(v_l)) ;                  // Simulation
6          Backpropagate(v_l, Δ) ;                // Backpropagation
7      end
8      return action of best child of v_0;
9  end
10 function TreePolicy(v)
11     while v is not a terminal node do
12         if v has unexpanded children then
13             return Expand(v);
14         end
15         else
16             v ← BestChild(v, C) ;   // Select child using (2.5)
17         end
18     end
19     return v;
20 end
21 function Expand(v)
22     Choose an untried action a from Actions(s(v));
23     Create a new child node v' with state s' = Result(s(v), a);
24     Add v' as a child of v;
25     return v';
26 end
27 function BestChild(v, C)
28     return arg max_{v'∈children(v)} W(v')/N(v') + C sqrt(ln N(v) / N(v'));
29 end
30 function Backpropagate(v, Δ)
31     while v is not null do
32         N(v) ← N(v) + 1;
33         W(v) ← W(v) + Δ ;                          // Q(v) = W(v)/N(v)
34         v ← parent(v);
35     end
36 end
```

2.3 Deep Reinforcement Learning

When the state and/or action spaces of an MDP are too large to be stored in a table, or when the environment dynamics $\mathcal{P}(s'|s,a)$ are unknown, we turn to model-free DRL. DRL combines RL with Deep Neural Network (DNN), using DNN as powerful function approximators to learn the policy or value functions directly from experience [17]. Generally, DRL can be categorized as *value-based* and *policy-based* methods.

2.3.1 Deep Q-Learning

Deep Q-Network (DQN) is a foundational value-based DRL algorithm focus on learning the optimal action-value function, $Q^*(s,a)$, by replacing the traditional Q-table with a neural network $Q(s,a;\theta)$, parameterized by weights θ. This allows it to handle high-dimensional state spaces, such as raw pixels from an image. To stabilize the learning process, DQN introduces two key innovations [41]:

1. **Experience Replay:** The agent stores its experiences, which is the transitions of (s,a,r,s'), in a large memory replay buffer $\mathcal{D}$. During training, mini-batches of experiences are randomly sampled from this buffer. This practice breaks the temporal correlations inherent in sequential observations, leading to more stable and efficient learning.

2. **Target Network:** A separate target network, $Q(s,a;\theta')$, with weights θ' that are periodically copied from the main online network $Q(s,a;\theta)$, is used to compute the TD target. This approach keeps the target values stable for a period of time, preventing the oscillations and divergence that can occur when a single network is used to both estimate the current value and the target value.

The neural network is trained by minimizing the Mean Squared Error (MSE) between its Q-value predictions and the TD target computed by the target network.

Algorithm 2: Double Deep Q-Network (DDQN) Algorithm

Data: Minibatch size B, sync frequency C, discount factor γ

`/* Initialization */`

Initialize replay buffer $\mathcal{D}$ with capacity N;

Initialize online Q-network Q with random weights θ;

Initialize target Q-network $\hat{Q}$ with weights $\theta' \leftarrow \theta$;

for *episode* $\leftarrow 1$ **to** M **do**

  Initialize the environment and get the initial state s_1;

  for *step* $t \leftarrow 1$ **to** T **do**

    `/* Select action using an` ϵ`-greedy policy */`

    if *a random value is less than* ϵ **then**

      Select a random action a_t;

    else

      Select action $a_t \leftarrow \arg\max_a Q(s_t, a; \theta)$;

    end

    `/* Execute action and store experience */`

    Execute action a_t, observe reward r_t, and get next state s_{t+1};

    Determine if the episode has terminated, done;

    Store the transition $(s_t, a_t, r_t, s_{t+1}, \text{done})$ in $\mathcal{D}$;

    `/* Sample minibatch and train online network */`

    Sample a random minibatch of B transitions $(s_j, a_j, r_j, s_{j+1}, \text{done}_j)$ from $\mathcal{D}$;

    Set the target value y_j for each transition in the minibatch:

$$y_j = \begin{cases} r_j & \text{if done}_j \\ r_j + \gamma \hat{Q}(s_{j+1}, \arg\max_{a'} Q(s_{j+1}, a'; \theta); \theta') & \text{otherwise} \end{cases}$$

    Perform a gradient descent step on the loss $\mathcal{L}(\theta)$ with respect to the online network weights θ:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_j \left(y_j - Q(s_j, a_j; \theta)\right)^2$$

    `/* Periodically update the target network */`

    Every C steps, copy the online network weights to the target network: $\theta' \leftarrow \theta$;

  end

end

The loss function is:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(\underbrace{r + \gamma \max_{a'} Q(s', a'; \theta')}_{\text{TD Target}} - Q(s, a; \theta)\right)^2\right]. \tag{2.6}$$

The overall process, which also incorporates experience replay and a target network, is detailed in Algorithm 2. Note that the original DQN algorithm suffers from an overestimation bias because it uses the same network to both select the best action and evaluate its value. Double DQN mitigates this issue by decoupling these two steps [41]. It uses the network $Q(s, a; \theta)$ to select the best next action, but uses the target network $\hat{Q}(s, a; \theta')$ to evaluate the value of that action. Hence, we present double DQN here as it has been recognized as the default implementation of single-agent deep Q-learning.

2.3.2 Soft Actor-Critic

Soft Actor-Critic (SAC) is a model-free off-policy reinforcement learning algorithm designed to maximize expected cumulative reward while simultaneously encouraging exploration through entropy maximization[42]. Its off-policy nature allows it to reuse past experiences stored in a replay buffer, leading to significant improvements in sample efficiency compared to on-policy actor-critic methods like Proximal Policy Optimization (PPO). The SAC algorithm is grounded in the maximum entropy reinforcement learning framework, where the objective is to optimize the policy π_θ by maximizing both the expected return and the entropy of the policy distribution:

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot|s_t))\right)\right], \tag{2.7}$$

where $\mathcal{H}(\pi(\cdot|s_t)) = -\sum_a \pi(a|s_t) \log \pi(a|s_t)$ denotes the entropy of the policy, and α is a temperature parameter that controls the trade-off between reward

maximization and entropy.

The Q-function $Q_\phi(s, a)$ is learned by minimizing the temporal difference (TD) error using a target network:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - y\right)^2\right], \tag{2.8}$$

with the target defined as:

$$y = r + \gamma\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[Q_{\bar{\phi}}(s',a') - \alpha\log\pi(a'|s')\right], \tag{2.9}$$

where $\bar{\phi}$ denotes the parameters of a slowly-updated target network.

The actor loss is derived from the policy improvement step of the soft policy iteration framework. The optimal policy minimizes the Kullback–Leibler (KL) divergence between the current policy and the energy-based distribution induced by the Q-function. This results in the following actor objective[43]:

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{s\sim\mathcal{D}}\left[\sum_a \pi_\theta(a|s)\left(\alpha\log\pi_\theta(a|s) - Q_\phi(s,a)\right)\right]. \tag{2.10}$$

This formulation encourages the policy to assign higher probability to actions with higher expected returns while maintaining sufficient entropy for exploration. The entropy term $\alpha\log\pi_\theta(a|s)$ penalizes certainty and thus prevents premature convergence to suboptimal deterministic policies. SAC alternates between updating the critic parameters to minimize the Bellman residual and updating the actor to minimize the above loss, ensuring policy improvement under the maximum entropy principle.

In practice, we adopt the twin critics trick to estimate stable Q-values. This can help mitigate the overestimation issue that is common in deep Q-learning [42]. Then, the loss for each critic $i \in \{1, 2\}$ is:

$$\mathcal{L}_{\text{critic}}(\phi_i) = \hat{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_{\phi_i}(s,a) - y(r,s')\right)^2\right], \tag{2.11}$$

Algorithm 3: Soft Actor-Critic (SAC) for Discrete Action Spaces

Initialize actor π_θ, critics Q_{ϕ_1}, Q_{ϕ_2}, target critics $Q_{\phi'_1}, Q_{\phi'_2}$ with $\phi'_i \leftarrow \phi_i$.
Initialize replay buffer $\mathcal{D}$;
for *each timestep* $t = 1, 2, \ldots$ **do**
  Sample action $a_t \sim \pi_\theta(\cdot|s_t)$;
  Execute a_t, observe reward r_t and next state s_{t+1};
  Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$;
  if *enough samples in* $\mathcal{D}$ **then**
    Sample a minibatch of transitions (s, a, r, s') from $\mathcal{D}$;
    `// Critic Updates`
    Compute the next state action probabilities $\pi_{\text{next}} \leftarrow \pi_\theta(\cdot|s')$;
    Compute the target Q-values: $Q'_{\text{next}} \leftarrow \min_{i=1,2} Q_{\phi'_i}(s')$;
    Compute the soft value target:
    $y \leftarrow r + \gamma \left(\pi^T_{\text{next}}(Q'_{\text{next}} - \alpha \log(\pi_{\text{next}}))\right)$;
    for $i = 1, 2$ **do**
      Update critic i by minimizing the MSE:
      $\mathcal{L}_{\phi_i} = \hat{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[(Q_{\phi_i}(s, a) - y)^2\right]$;
    end
    `// Actor and Alpha Updates`
    Compute current action probabilities $\pi_{\text{current}} \leftarrow \pi_\theta(\cdot|s)$;
    Compute current Q-values: $Q_{\text{current}} \leftarrow \min_{i=1,2} Q_{\phi_i}(s)$;
    Update actor by minimizing:
    $\mathcal{L}_\theta = \hat{\mathbb{E}}_{s\sim\mathcal{D}}\left[\pi^T_{\text{current}}(\alpha \log(\pi_{\text{current}}) - Q_{\text{current}})\right]$;
    (Optional) Update temperature α by minimizing
    $\mathcal{L}_\alpha = \hat{\mathbb{E}}_{s\sim\mathcal{D}}\left[-\alpha(\log \pi_\theta(s) + \bar{\mathcal{H}})\right]$;
    Update target networks: $\phi'_i \leftarrow \tau\phi_i + (1 - \tau)\phi'_i$ for $i = 1, 2$;
  end
end

where the target $y(r, s')$ is given by:

$$y(r, s') = r + \gamma \left(\pi_\theta(s')^T \left(\min_{i=1,2} Q_{\text{target},i}(s') - \alpha \log \pi_\theta(s')\right)\right). \tag{2.12}$$

Analogously, the actor's loss function is given by:

$$\mathcal{L}_{\text{actor}}(\theta) = \hat{\mathbb{E}}_{s\sim\mathcal{D}}\left[\pi_\theta(s)^T(\alpha \log \pi_\theta(s) - \min_{i=1,2} Q_{\phi_i}(s))\right]. \tag{2.13}$$

To stabilize learning, the temperature α can also be tuned automatically via dual gradient descent to match a target entropy [43]. Note that SAC is most commonly used for continuous actions, though it can theoretically support both continuous and discrete actions. When adapted for discrete actions, the policy is represented a categorical distribution over actions and the critics should output estimates over all possible actions [44]. The full procedure for SAC with discrete actions is detailed in Algorithm 3.

2.4 Multi-Agent Reinforcement Learning

MARL extends the single-agent RL paradigm to settings with multiple interacting agents. This extension is crucial for addressing a wide range of decentralized decision-making problems in modern communication systems, such as the distributed packet routing in LEO satellite networks that is a focus of this dissertation [37]. In a multi-agent system, the environment's dynamics and the rewards received by an agent depend not only on its own actions but on the joint actions of all agents. This interdependency introduces significant complexities not present in the single-agent case.

2.4.1 The Dec-POMDP Framework

Multi-agent environments are often formally modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). A Dec-POMDP for N agents is a tuple $\langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \mathcal{P}, \{r_i\}_{i=1}^N, \{\Omega_i\}_{i=1}^N, \mathcal{O}, \gamma \rangle$, where:

- $\mathcal{S}$ is the global state space.
- $\mathcal{A}_i$ is the action space for agent i. The joint action space is $\mathcal{A} = \times_{i=1}^N \mathcal{A}_i$.
- $\mathcal{P}(s'|s, \mathbf{a})$ is the transition function, given the global state s and joint action $\mathbf{a} = (a_1, \ldots, a_N)$.

- $r_i(s, \mathbf{a})$ is the reward function for agent i.
- Ω_i is the observation space for agent i.
- $\mathcal{O}(o_i|s, \mathbf{a})$ is the observation function, giving the probability of agent i receiving local observation o_i from global state s after the joint action $\mathbf{a}$.

Each agent i learns a local policy $\pi_i(a_i|o_i)$ based only on its own observation history. The goal is typically to find a set of policies that maximizes a common objective (cooperative setting) or reaches a stable equilibrium (competitive or mixed setting).

2.4.2 Fundamental Challenges in MARL

The transition from a single-agent to a multi-agent setting introduces several fundamental challenges that complicate the learning problem [45]. These challenges are the primary drivers for the development of specialized MARL algorithms.

Non-Stationarity

From the perspective of any single agent, the environment appears non-stationary. As other agents are simultaneously learning and updating their policies, the dynamics of the environment from one agent's viewpoint are constantly changing. An action that was once optimal may become suboptimal as other agents adapt their behaviors. This violates the stationary Markov assumption that supports the convergence guarantees of many single-agent RL algorithms, making learning unstable and convergence difficult to achieve.

Scalability and the Curse of Dimensionality

Another major challenge is scalability. In a system with N agents, the joint action space grows exponentially with the number of agents. For example, if each of the N agents has $|\mathcal{A}|$ discrete actions, the joint action space has a size of $|\mathcal{A}|^N$.

This *curse of dimensionality* makes it computationally intractable for any agent to explicitly model the actions of all other agents or for a centralized controller to explore the entire joint action space. This issue necessitates the use of scalable approaches that do not require explicit enumeration of joint actions.

Partial Observability

In most realistic decentralized systems, agents operate with only partial information about the global state. An agent typically has access to its own local observations, which may be noisy or incomplete representations of the true system state. This partial observability, formalized by the Dec-POMDP framework, makes it difficult for an agent to infer the intentions or policies of other agents, further exacerbating the non-stationarity and credit assignment problems.

2.4.3 MARL Paradigms & Implementations

To address these challenges, several learning paradigms have been developed. The choice of paradigm often involves a trade-off between scalability, learning stability, and performance.

Independent Learners

The most straightforward approach to MARL is to allow each agent to learn independently, treating all other agents as part of the environment. This paradigm is known as Independent Q-Learning (IQL) or Independent Actor-Critic (IAC). In this setup, each agent i maintains its own policy $\pi_{\theta_i}(a_i|o_i)$ and/or value function $Q_{\phi_i}(o_i, a_i)$ and updates its parameters using only its local experiences (o_i, a_i, r_i, o_i'). The general procedure for an independent actor-critic agent is outlined in Algorithm 4. While we present its pseudo code in a synchronized way, independent MARL can surely support Decentralized Training with Decentralized Execution (DTDE) and work in an asynchronous manner.

Algorithm 4: Independent Actor-Critic (IAC)

Initialize actor π_{θ_i} and critic Q_{ϕ_i} for each agent $i \in \{1, \dots, N\}$;
Initialize replay buffer $\mathcal{D}_i$ for each agent i;
for *each episode* **do**
  Receive initial observation $o_1, \dots, o_N$;
  for *each timestep* $t = 1, 2, \dots, T$ **do**
    For each agent i, sample action $a_{i,t} \sim \pi_{\theta_i}(\cdot|o_{i,t})$;
    Execute joint action $\mathbf{a}_t = (a_{1,t}, \dots, a_{N,t})$;
    Observe rewards $r_{1,t}, \dots, r_{N,t}$ and next observations $o'_{1,t}, \dots, o'_{N,t}$;
    For each agent i, store transition $(o_{i,t}, a_{i,t}, r_{i,t}, o'_{i,t})$ in $\mathcal{D}_i$;
    if *enough samples in* $\mathcal{D}_i$ **then**
      Sample a minibatch from $\mathcal{D}_i$;
      Update critic ϕ_i by minimizing its local TD error;
      Update actor θ_i using the policy gradient with its local critic's values;
    end
  end
end

While simple and highly scalable, this simplicity comes at a cost. It suffers from a critical theoretical challenge: **non-stationarity**. From the perspective of any individual agent, the environment appears to be non-stationary because the other agents' policies are simultaneously changing during the learning process. This violates the Markov assumption that supports the convergence guarantees of most single-agent RL algorithms [45]. As a result, it is possible that independent learners fail to coordinate effectively and may not converge to a stable or optimal joint policy. Despite these theoretical limitations, independent learning can be surprisingly effective in practice and serves as a very strong baseline [46].

Cooperative Learners & Synchronization

To address the non-stationarity issue in a more principled way, cooperative MARL algorithms often adopt the paradigm of Centralized Training with Decentralized Execution (CTDE). The core idea is to leverage global information during the

training phase to stabilize learning, while ensuring that the resulting policies can be executed in a decentralized manner using only local information.

Under the CTDE framework, a centralized critic is trained, which has access to the global state or the joint observations and actions of all agents. This critic learns a joint action-value function, $Q(\mathbf{o}, \mathbf{a})$, where $\mathbf{o} = (o_1, \ldots, o_N)$ and $\mathbf{a} = (a_1, \ldots, a_N)$. By conditioning on the global information, the critic provides a stable learning signal for the actors, effectively resolving the non-stationarity problem. Concurrently, each agent i maintains a decentralized actor, $\pi_{\theta_i}(a_i|o_i)$, which is trained to maximize the expected return predicted by the centralized critic. The general synchronous CTDE training loop is shown in Algorithm 5.

Algorithm 5: Cooperative MARL using synchronous CTDE

Initialize decentralized actors π_{θ_i} for each agent $i \in \{1, \ldots, N\}$;
Initialize centralized critic $Q_\phi(\mathbf{o}, \mathbf{a})$;
Initialize a shared replay buffer $\mathcal{D}$;
for *each episode* **do**
  Receive initial joint observation $\mathbf{o}_1 = (o_{1,1}, \ldots, o_{N,1})$;
  for *each timestep* $t = 1, 2, \ldots, T$ **do**
    `// Decentralized Execution`
    For each agent i, sample action $a_{i,t} \sim \pi_{\theta_i}(\cdot|o_{i,t})$;
    Execute joint action $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t})$;
    Observe shared reward r_t and next joint observation $\mathbf{o}'_t$;
    Store joint transition $(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}'_t)$ in $\mathcal{D}$;
    `// Centralized Training`
    if *enough samples in* $\mathcal{D}$ **then**
      Sample a minibatch of joint transitions from $\mathcal{D}$;
      Update centralized critic ϕ by minimizing the joint TD error;
      For each agent i, update actor θ_i using the policy gradient derived from the centralized critic;
    end
  end
end

While CTDE provides a robust solution to non-stationarity, it introduces its own challenges. The centralized critic itself introduces a significant **scalability**

bottleneck. The input to the critic is the joint observation and action space of all agents, which can become prohibitively large as the number of agents increases. For instance, in a system with many agents, such as a LEO constellation with more than 1000 satellites, the input dimension for the critic's neural network can become massive, making it difficult to train effectively. An extremely wide or deep network can be problematic, causing substantial degradation in training stability and sample efficiency [47]. This scalability issue with the centralized critic limits the applicability of standard CTDE methods in large-scale systems and motivates research into more scalable architectures, such as those employing value-decomposition or parameter sharing [45].

Furthermore, most CTDE implementations assume that agents act in synchronized discrete time steps. This lockstep execution model, illustrated in Fig. 2.3, is

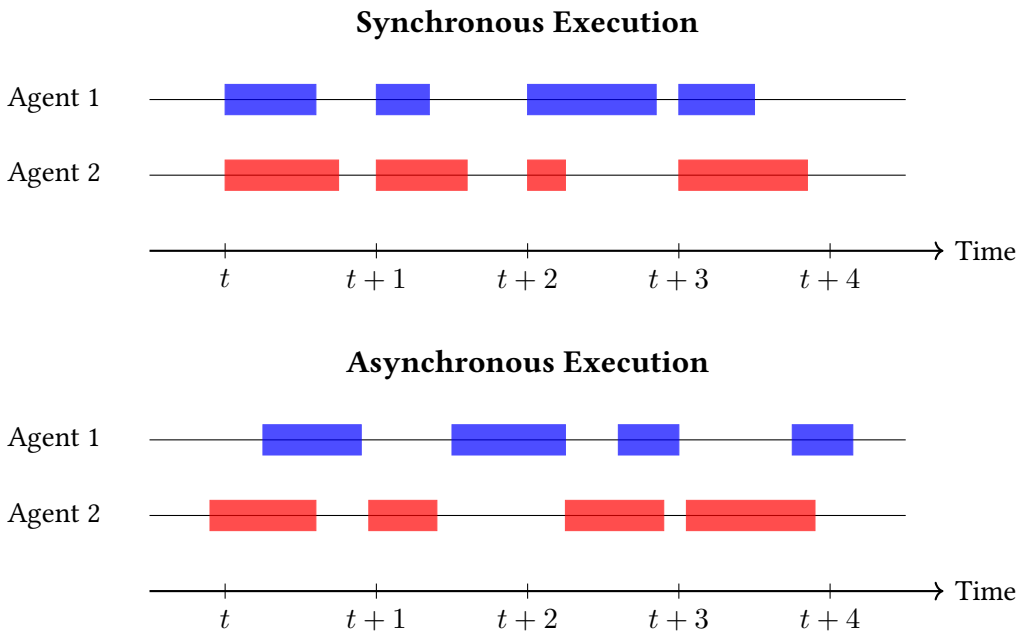

Figure 2.3 Comparison of multi-agent action execution models. **Top (Synchronous):** Agents operate in a synchronized *lockstep*, starting actions only at discrete time ticks $(t, t+1, \dots)$. This rigid turn-based model requires agents that finish early to wait for others. **Bottom (Asynchronous):** Agents act independently of a global clock. Actions can start at any time and have variable durations, reflecting a more flexible and realistic scenario.

often impractical for real-world large-scale communication systems [47, 48, 49, 50]. For example, in LEO networks, packet routing is an inherently event-driven and asynchronous process. Packet arrivals and departures occur at continuous and unpredictable times, which requires satellites to make routing decisions independently based on their local event timelines (paces). Forcing this naturally asynchronous process into a rigid synchronous joint-action model, as shown in the top panel of Fig. 2.3, introduces artificial delays and severe scalability bottlenecks, as all agents must wait for a *global tick* before acting [47]. This impractical assumption of action-synchronization highlights a critical gap in existing research.

Although coordination can be encouraged during training via a centralized critic and forced synchronization (action padding), this setup often fails to reflect the real-world asynchronous execution environment [49]. As a result, the training-execution mismatch between synchronized training and decentralized event-driven execution can lead to fragile coordination and significant performance degradation in practice [49, 50]. While independent learning frameworks (as discussed previously) also operate asynchronously, their complete lack of centralized training leads to foreseeable weak coordination and performance loss. This highlights the need for a framework that is both asynchronous and capable of managing the risks that arise from uncoordinated decentralized actions.

Therefore, a core motivation of this dissertation is to develop an asynchronous MARL framework that eliminates unrealistic synchronization assumptions, allowing each agent to operate on its own event-driven timeline while still achieving robust self-coordinated behavior.

Part I

Model-Based Risk-Aware Learning with World Models

3 Risk-Aware Antenna Selection for Massive MU-MIMO under Incomplete CSI

Prologue. This chapter introduces the first of two core paradigms for the unified risk-aware decision-making frame developed in this dissertation: *model-based risk-aware planning*. As established in the introduction, this paradigm is suited for scenarios where an underlying model of the system's dynamics can be learned from data, even when real-time observations are incomplete. We instantiate this approach in the context of a critical problem in 6G terrestrial networks: antenna selection in massive MIMO with incomplete CSI. This application serves as an ideal testbed, as the challenge of operating with incomplete CSI is a canonical example of *partial observability at scale*.

This chapter will detail the *Joint Channel Prediction and Antenna Selection (JCPAS)* framework, which integrates a predictive *"world model"* to estimate full CSI from partial measurements with a novel decision-making planner. The core contribution is the *Risk-Aware Monte-Carlo Tree Search (RA-MCTS)*, a planning algorithm that explicitly manages the risk of QoS violations by reasoning over the uncertainty inherent in the predicted channel states. In doing so, this chapter, based on the work published in [32], lays the foundation for the model-based methodology of Part I and provides a concrete solution to the problem of making robust decisions under channel uncertainty and partial observability.

3.1 Introduction

Massive multiple-input multiple-out (MIMO) has became the key technology to support the continuous development of future wireless network applications [8, 51, 52, 53]. By deploying a very large number of antennas at the base station (BS), massive MIMO is capable of significantly improving the spectral efficiency via spatial multiplexing gain [52]. However, the full potential of massive MIMO requires a huge number of dedicated radio frequency (RF) chains for every antenna, which results in not only increased capital expenditure (CAPEX) but also higher system energy consumption [51, 52, 53]. In practical massive MIMO systems, it is more cost-effective and energy-efficient to employ a number of RF chains less than the number of antennas, while the full spatial multiplexing gain can be preserved via antenna selection (AS) [52, 53]. Being a key component of hybrid signal processing techniques, AS aims to select the best subset of antennas for data transmission to reduce hardware cost and power consumption without losing the full potentials of antenna arrays [51]. During the last decades, the AS problem has been extensively studied in the presence of complete channel state information (CSI) (sometimes refer to full CSI), and it has been shown that AS can provide similar spectral efficiency with a lower energy consumption compared to the case without AS [52, 53].

3.1.1 Antenna Selection in Massive MIMO

In principle, the AS problem can be formulated as an integer programming problem under the assumption of complete CSI acquisition [53, 54]. The problem is NP-hard, thus solving it optimally imposes a prohibitive computational complexity which exponentially grows with the number of antennas in the worst case[54]. AS under complete CSI has been extensively investigated by researchers with the objective of designing a near-optimal algorithm with low computational complexities for massive MIMO [55, 30, 56, 54, 25, 57, 58, 59, 60]. However, it

should be noted that most existing works were established in the presence of complete CSI, which implies that the channel coefficients of all antennas must be fully observed [31, 61, 19].

Unfortunately, this assumption does not strictly hold for practical scenarios, especially with very large-scale antenna arrays [62]. This is mainly because complete CSI acquisition turns out to be a time-consuming task in such scenarios. In practice, CSI acquisition is accomplished by pilot sequences which consume radio resources proportional to the numbers of active antennas and users. Since the number of RF chains is smaller than the number of antennas, only an incomplete CSI can be observed and estimated from each pilot transmission. This indicates that acquiring complete CSI eventually leads to extra pilot transmissions and reducing the effective transmission rate [31, 19, 63, 62]. For instance, acquiring complete CSI could occupy more than half of the frame duration for the downlink of a typical time-division duplexing (TDD) massive MIMO system with 64 antennas, 16 RF chains and 4 single-antenna users [64, 31]. Therefore, it is of importance to take incomplete CSI acquisition into account in practical AS design, especially for very large antenna array configurations.

Antenna Selection with Channel Prediction

In order to address this issue, one promising approach would be the joint use of channel prediction and AS. It is straightforward that the extra pilot overhead can be reduced if we are able to predict the complete channel states. This approach works mainly because the real propagation environment is often time-varying and temporally correlated due to Doppler effects [65, 66, 67]. Based on this, researchers have developed different channel prediction algorithms in recent years [68, 69, 70]. In [68], the authors proposed a spatio-temporal autoregressive method for the prediction of the high mobility channel, where the prediction was performed utilizing the temporal correlation in the angle-delay domain. In [69], the authors predicted channel states by exploiting the channel correlations.

The proposed method employed the convolutional neural network (CNN) and the long-short-term memory network (LSTM), which allows a multistep prediction of the channels. Similarly in [70], the authors predicted channel states by utilizing the spatio-temporal characteristics of CSI and a combination of CNN and convolutional LSTM. Although showing improved AS performance, these channel prediction methods are based on the historic complete CSI measurements [68, 69, 70]. It is noted that obtaining complete CSI is spectral-inefficient in the considered massive MIMO systems, especially when the number of antennas significantly exceeds the number of RF chains. In addition, the aforementioned channel predictions are deterministic methods, which limits the use of channel statistics for performance enhancement.

Antenna Selection with Incomplete CSI

Recently, researchers have focused on developing AS algorithms which can directly operate under incomplete CSI [31, 19, 63]. In [31], the authors formulated the AS problem as a combinatorial multi-armed bandit problem when only incomplete CSI is available, and proposed an online AS algorithm using Thompson sampling. However, it is assumed therein that each antenna contributes equally to the sum capacity. Since this assumption is not valid in most real propagation environments [62], the solution therein is not robust in practical scenarios. The authors of [19] considered the AS as a partially observable Markov decision process (POMDP) and proposed a myopic policy for selecting antennas under imcomplete CSI. The myopic policy maintains a belief vector for the underlying channel states of each time slot, and updates this belief vector along with the system dynamics. The myopic policy therein, however, was only designed for single-user MIMO systems under general fading channels with a two-state coarse channel quantization. Since practical quality-of-service (QoS) constraints were not considered in [31, 19], their applicability is limited in practical massive MIMO systems.

Although several approaches have been proposed to address the antenna selection problem under incomplete CSI, further improvements are still needed to improve system performance and robustness. For the practical AS algorithm design, one crucial concern is how to maximize the system performance while reducing the chance of violating the system' practical constraints when only incomplete CSIs are available. Risk-aware solutions using conditional value at risk (CVaR) have recently been proposed for resource management in ultra-reliable and low latency communications (URLLC) and the coexistence of eMBB and URLLC services [23, 71, 72]. However, these methods rely on the complete information of the system states, which are not applicable in the considered scenario.

3.1.2 Motivations and Contributions

As mentioned above, existing approaches cannot efficiently solve the antenna selection problem in multi-user massive MIMO and incomplete CSI. This motivates us to design a general antenna selection framework that can operate robustly against the complete CSI condition. Additionally, existing AS algorithms lack the capability to recognize the risk of violating the system constraints under incomplete CSI, which is essential to the required QoS. In general, risk awareness should be presented throughout the decision process, which implies that a desired antenna selection should be performed by jointly considering three important factors: practical system constraints, optimization objectives, and the uncertainties introduced by the incomplete CSI. Therefore, risk-aware planning remains a challenging issue for practical AS solutions, which is also the motivation of this work.

In this chapter, we propose a joint channel prediction and antenna selection framework (JCPAS) for the antenna selection problem in multi-user massive MIMO under incomplete CSI and practical system constraints. The proposed JCPAS comprises a deep unsupervised learning-based conditional channel estimator

and a risk-aware Monte Carlo tree search (RA-MCTS) algorithm. A risky event is identified when one of the system constraints cannot be satisfied. At each frame, the channel estimator maintains a belief distribution by estimating conditional channel statistics from the sequence of the past incomplete CSI measurements and estimates the posterior channel distribution. Based on the estimated posterior channel distribution, the RA-MCTS algorithm evaluates uncertain outcomes of each possible antenna combination through Monte Carlo simulations. In such a risk-aware manner, the chance of violating the system constraints can be reduced, and the corresponding negative consequences can also be mitigated when arisen. Simulation results show that the proposed RA-MCTS algorithm not only cuts the average power consumption by 50%, but also significantly reduces the probability of violating the system constraints by 90%.

To summarize, our main contributions are as follows:

- We introduce the JCPAS framework in massive MIMO systems under incomplete CSI. The proposed JCPAS framework does not require complete CSI measurements and is robust to conventional antenna selection algorithms. In addition, the proposed channel prediction method is a probabilistic model that can be used to enhance the performance of the integrated selection algorithm.

- We propose the RA-MCTS algorithm which enables efficient and robust antenna selection in massive MIMO under incomplete CSI and practical system constraints. In contrast to the existing antenna selection algorithm, our proposed RA-MCTS is applicable to diverse optimization objectives and system constraints, and it is able to reduce the chance of violating the practical system constraints by leveraging channel statistics.

- We provide a new insight for risk-aware decision making with limited resources and insufficient information. In particular, a risk-aware system can be built by leveraging the historical incomplete observations to estimate

a belief distribution over the underlying system dynamics and planning based on the statistics of the random outcomes introduced by incomplete observations accordingly.

The remainder of this paper is organized as follows. Section II describes the mathematical model of the considered massive MU-MIMO system, as well as the associated antenna selection problem. Section III presents the proposed deep unsupervised learning-based conditional channel estimator. Section IV presents the RA-MCTS algorithm, which is a risk-aware planning algorithm for selecting antennas with incomplete observations. Section V discusses the related simulation results. Finally, Section VI concludes the paper.

3.2 Preliminaries

In this section, we introduce the system model of the considered massive MIMO system as well as the associated AS problem. After that, we will review the greedy search AS algorithm under the complete CSI assumption.

3.2.1 System Model

As shown in Fig. 3.1, we consider the downlink of a massive MU-MIMO system where a BS serves N_u single-antenna users. The BS is equipped with N_t transmit antennas and N_f ($0 < N_f \ll N_t$) RF chains. In addition, switches are also available at the BS such that an RF chain can connect with any antenna of interest. The channel between the BS and users is time-varying and temporally correlated, which is common in real propagation environments with Doppler effects [65, 66, 67]. Moreover, we assume that the CSI remains unchanged within each frame duration of T channel uses (c.u.), and the considered system operates in TDD mode, meaning that we have identical channels for both uplink and downlink transmission due to channel reciprocity [56, 54]. On the downlink transmission,

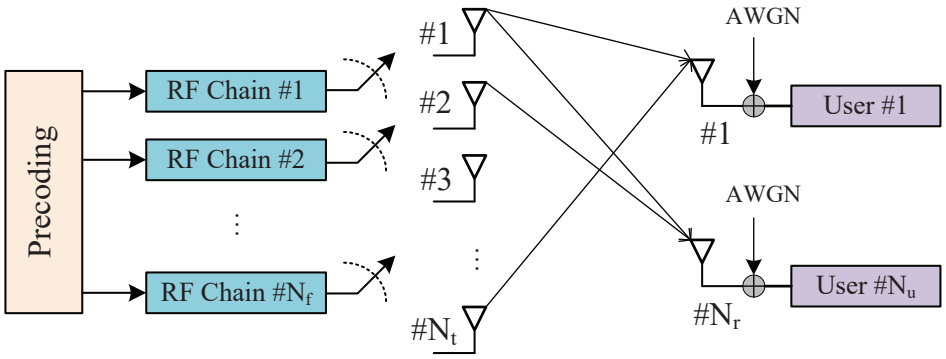

Figure 3.1 Structure of the considered massive MIMO system.

the CSI acquisition is accomplished via uplink pilot-assisted channel measurement, and multi-user precoding is then adopted to mitigate inter-user interference. Under these settings, we further denote τ_{csi} as the number of c.u. consumed to acquire CSI, resulting in $T - \tau_{csi}$ c.u. for data transmission.

Since only N_f out of N_t transmit antennas can be activated at the same time, the BS needs to select the best N_f antennas in terms of performance metrics maximization. Let $\boldsymbol{a} = \{a_1, a_2, \ldots, a_j, \ldots, a_{N_f}\}$ be the set of the indices of the N_f selected antennas, and we denote $\mathcal{A}$ as the set of all possible antenna combinations with a cardinality of $|\mathcal{A}| = \binom{N_t}{N_f}$. In addition, let $\boldsymbol{H} \in \mathbb{C}^{N_u \times N_t}$ be the complete CSI matrix, and we denote $\boldsymbol{H}(\boldsymbol{a})$ as the incomplete (or partial) CSI from the chosen combination $\boldsymbol{a} \in \mathcal{A}$, meaning that the columns of $\boldsymbol{H}(\boldsymbol{a})$ are selected from $\boldsymbol{H}$ with respect to the indices in $\boldsymbol{a}$. Besides, we denote x_k as the data symbol to be transmitted to user k, and $\mathbb{E}\{|x_k|^2\} = 1$. Then, the received signal $y_k(\boldsymbol{a})$ at user k is given by

$$y_k(\boldsymbol{a}) = \boldsymbol{h}_k(\boldsymbol{a})_k \boldsymbol{w}_k x_k + \sum\nolimits_{j \neq k} \boldsymbol{h}_j(\boldsymbol{a}) \boldsymbol{w}_j x_j + n_k, \tag{3.1}$$

where $\boldsymbol{h}_k(\boldsymbol{a}) \in \mathbb{C}^{1 \times N_f}$ is the channel vector for user k from the antenna combination $\boldsymbol{a}$, $\boldsymbol{w}_k$ denotes the $N_f \times 1$ precoding vector for user k, and $n_k \sim \mathcal{CN}(0, \sigma_k^2)$ is the additive white Gaussian noise (AWGN) at user k. The second term of (3.1) is the inter-user interference at user k.

Assuming negligible processing time, the *effective spectral efficiency* for downlink transmission to user k with the selected antennas $\boldsymbol{a}$ can be written as

$$R_k(\boldsymbol{a}) = \left(1 - \frac{\tau_{csi}}{T}\right) \log_2 \left(1 + \text{SINR}_k(\boldsymbol{a})\right), \tag{3.2}$$

where $\text{SINR}_k(\boldsymbol{a}) = \frac{\|\boldsymbol{h}_k(\boldsymbol{a})\boldsymbol{w}_k(\boldsymbol{a})\|^2}{\sigma_k^2 + \sum_{j \neq k} \|\boldsymbol{h}_k(\boldsymbol{a})\boldsymbol{w}_j(\boldsymbol{a})\|^2}$ represents the signal-to-interference-noise ratio (SINR) of user k. Accordingly, the *effective sum spectral efficiency* with antenna combination $\boldsymbol{a}$ can be bounded by $C(\boldsymbol{a}) = \sum_{k=1}^{N_u} R_k(\boldsymbol{a})$, and the total power consumption for transmitting data at each frame is given by $P(\boldsymbol{a}) = \sum_{k=1}^{N_u} \|\boldsymbol{w}_k(\boldsymbol{a})\|^2$.

3.2.2 Antenna Selection with Objective Maximization

For practical scenarios, a typical objective for selecting the best subset of antennas is to optimize a generic objective function $\mathcal{F}_k(\boldsymbol{a})$ under the constraints of total transmit power and minimum QoS requirements. Mathematically, the optimization problem can be formulated as

$$\underset{\boldsymbol{a} \in \mathcal{A}}{\text{maximize}} \;\; \mathcal{F}(\boldsymbol{a}), \tag{3.3}$$

$$\text{subject to} \;\; P(\boldsymbol{a}) \leq P_{tot}, \tag{3.4}$$

$$R_k(\boldsymbol{a}) \geq \eta_k, \forall k, \tag{3.5}$$

where η_k is the QoS requirement for user k, P_{tot} denotes the total transmit power, and $\mathcal{F}(\boldsymbol{a})$ is the objective function of interest. According to the specific problem, the objective function can be $\mathcal{F}(\boldsymbol{a}) = -\sum_{k=1}^{N_u} \|\boldsymbol{w}_k(\boldsymbol{a})\|^2$ when we want to minimize the energy consumption, and $\mathcal{F}(\boldsymbol{a}) = \sum_{k=1}^{N_u} R_k(\boldsymbol{a})$ if we want to maximize the sum-throughput for the system.

Antenna Selection with Complete CSI

Solutions to problem (3.3) have been well studied in the literature under the *complete CSI* assumption [25, 57, 58, 59, 60], that it is possible to fully observe the channel states with an affordable overhead. Among these solutions, the idea of greedy search is widely adopted due to its good performance and low-complexity [54]. Let $\boldsymbol{a}^p$ be a set of p selected antenna indices, and let $\boldsymbol{a}^q \supset \boldsymbol{a}^p$ be a superset of $\boldsymbol{a}^p$ with $q > p$. In addition, we denote $\boldsymbol{a}^{q-p} = \boldsymbol{a}^q \setminus \boldsymbol{a}^p$ as the difference between set $\boldsymbol{a}^q$ and $\boldsymbol{a}^p$. According to the Proposition 1 of [56], the spectral efficiency loss of removing $q-p$ antennas from $\boldsymbol{a}^q$ can be bound by

$$C(\boldsymbol{a}^{q-p}) = \log_2\Big(1 + \frac{P_{tot} \cdot \mathrm{tr}(\boldsymbol{\Lambda}_{q-p})}{\mathrm{tr}(\boldsymbol{Q}_q)^2 + \mathrm{tr}(\boldsymbol{Q}_q)\,(P_{tot} + \mathrm{tr}(\boldsymbol{\Lambda}_{q-p}))}\Big),$$

with the following notations

$$\boldsymbol{\Lambda}_{q-p} = \boldsymbol{Q}_q \boldsymbol{H}(\boldsymbol{a}^{q-p}) \boldsymbol{A}_{q-p}^{-1} \boldsymbol{H}(\boldsymbol{a}^{q-p})^H \boldsymbol{Q}_q, \tag{3.6}$$

$$\boldsymbol{A}_{q-p} = (\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{a}^{q-p})^H \boldsymbol{Q}_q \boldsymbol{H}(\boldsymbol{a}^{q-p})), \tag{3.7}$$

$$\boldsymbol{Q}_q = \Big(\boldsymbol{H}(\boldsymbol{a}^q)\boldsymbol{H}(\boldsymbol{a}^q)^H\Big)^{-1}, \tag{3.8}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace. Now, the sum-throughput maximization problem can be converted as removing $N_t - N_f$ antennas with the minimum capacity loss, which is given by

$$\boldsymbol{a}^* = \underset{\boldsymbol{a} \in \mathcal{A}}{\arg\min}\, \mathrm{tr}(\boldsymbol{\Lambda}_{N_t - N_f}). \tag{3.9}$$

Unfortunately, the above equation still needs exhaustive search to find the best antenna combination to be removed. Nevertheless, it is possible to reduce the computational complexity by utilizing the concept of greedy search. Then, each time the greedy search suggests to remove the worst antenna that contributes least to the capacity, resulting in that the search space can be reduced significantly

Algorithm 6: Greedy Search Algorithm

Input: Full channel matrix $\boldsymbol{H}$
Output: Selected antenna combination $\boldsymbol{a}$

Let the set of all antennas be $\boldsymbol{\mathcal{N}} = \{1, \ldots, N_t\}$;
Initialize the set of removed antennas as $\boldsymbol{r} \leftarrow \emptyset$;
Define $\boldsymbol{A} \leftarrow (\boldsymbol{H}\boldsymbol{H}^H)^{-1}$;
while $|\boldsymbol{r}| < N_t - N_f$ **do**
  $m \leftarrow \arg\min_{i \in \boldsymbol{\mathcal{N}} \setminus \boldsymbol{r}} \frac{\left\|\boldsymbol{h}_i^H \boldsymbol{A}\right\|^2}{1 - \boldsymbol{h}_i^H \boldsymbol{A} \boldsymbol{h}_i}$;
  $\boldsymbol{A} \leftarrow \boldsymbol{A} + \frac{\boldsymbol{A}\boldsymbol{h}_m \boldsymbol{h}_m^H \boldsymbol{A}}{1 - \boldsymbol{h}_m^H \boldsymbol{A} \boldsymbol{h}_m}$;
  $\boldsymbol{r} \leftarrow \boldsymbol{r} \cup \{m\}$;
end
$\boldsymbol{a} \leftarrow \boldsymbol{\mathcal{N}} \setminus \boldsymbol{r}$;
return $\boldsymbol{a}$;

while still reserving a good performance. According to Proposition 2 of [56], the antenna to be removed at each iteration is given by

$$m = \arg\min_{r \in \boldsymbol{a}^p} \frac{\left\| \boldsymbol{h}_r^H \left(\boldsymbol{H}(\boldsymbol{a}^p) \boldsymbol{H}(\boldsymbol{a}^p)^H \right)^{-1} \right\|^2}{1 - \boldsymbol{h}_r^H \left(\boldsymbol{H}(\boldsymbol{a}^p) \boldsymbol{H}(\boldsymbol{a}^p)^H \right)^{-1} \boldsymbol{h}_r}, \tag{3.10}$$

where $\boldsymbol{a}^p$ is currently selected antenna set. By repeating this strategy, the antenna combination which maximizes the sum-throughput will eventually be determined [56], and the pseudocodes of the resulting greedy search algorithm is shown in Algorithm 6. It is worth noting that Algorithm 1 is directly applied to generalized ZF by using $\boldsymbol{A} = (\alpha \boldsymbol{I} + \boldsymbol{H}\boldsymbol{H}^G)^{-1}$.

Antenna Selection with Incomplete CSI

Due to the limited number of RF chains at BS, only the partial CSI corresponding to the N_f selected antennas could be measured in each pilot transmisison. In this case, we need $\tau_{csi} = N_u \lceil \frac{N_t}{N_f} \rceil$ c.u. to acquire the complete CSI [31], and obviously, τ_{csi} will quickly become cost-prohibitive for large N_t, where extra pilot overhead grows rapidly in massive MIMO and results in a reduced effective transmission

rate, as shown in (3.2). For this reason, acquiring complete CSI is thereby a very inefficient strategy for massive MIMO systems, which motivates us to study the antenna selection problem in the presence of incomplete (or partial) CSI.

3.3 Proposed Antenna Selection Framework with Incomplete CSI

In this section, we propose a joint channel prediction and antenna selection (JCPAS) framework operating without relying on a complete CSI assumption. JCPAS consists of two main blocks: *Channel Prediction* and *Antenna selection*. The first block learns the belief of the complete channel matrix from historical incomplete channel estimates (of selected antennas) and predicts the complete channel matrix. Based on the predicted complete channel matrix, the second block selects the best antenna subset.

3.3.1 Channel Prediction with Incomplete CSI History

In practical scenarios with large-scale antenna configurations, incomplete measurements of the current channel state are expected due to the limited number of RF chains and channel estimation duration. This limitation indicates that the antenna selection will be performed in the presence of incomplete channel information, in which existing complete CSI based antenna selection algorithms cannot be efficiently applied. Therefore, it is of vital importance to track the transition of channel states by exploiting the temporal correlation of real propagation environments, and thereby maintaining an accurate belief on the current channel state to help AS. We note that Kalman filtering based prediction is not applicable since it requires the complete historic CSI measurements.

From this perspective, we hereby propose a world model based on deep conditional generative model (DCGM), for estimating the distribution of the belief

state based on the sequence of past incomplete channel measurements. This model will serve as the *"world model"* for the massive MU-MIMO system for understanding the underlying channel dynamics. Mathematically, the channel prediction model can be considered as a probabilistic generative model $\mathcal{P}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$, which is conditioned on the history of incomplete CSI measurements, denoted by $\boldsymbol{\Phi}_t \triangleq \{\boldsymbol{H}_{t-L+1}(\boldsymbol{a}_{t-L+1}), \ldots, \boldsymbol{H}_{t-1}(\boldsymbol{a}_{t-1}), \boldsymbol{H}_t(\boldsymbol{a}_t)\}$ with a finite horizon L. In general, the value of L is determined by the storage and computing resources. Note that the missing entries will be replaced by zero entries in the sequence if the current time slot $t < L$. Although it is hard to estimate the exact posterior distribution $\mathcal{P}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$, we can still get its accurate approximation by the maximum likelihood approximation.

To this end, we begin by denoting the approximate distribution as $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ with parameters $\boldsymbol{\theta}$. Then, we train the model with the objective of maximizing the likelihood of the training samples on the chosen distribution, which is given by

$$\mathcal{N}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N_s} \sum\nolimits_{t=1}^{N_s} -\log \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t), \tag{3.11}$$

where N_s denotes the size of the training sample set $\mathcal{D} = \{\boldsymbol{H}_t, \boldsymbol{\Phi}_t\}_{t=1}^{N_s}$. Clearly, (3.11) quantifies the magnitude of fitness between the chosen distribution $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ and the samples drawn from the real distribution $\boldsymbol{H}_t \sim \mathcal{P}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$. In particular, $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ accurately approximates $\mathcal{P}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ when (3.11) achieves its minimum. On the other hand, $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ deviates from the real distribution when (3.11) enlarges. However, prior knowledge on the real distribution is needed to select an appropriate model for approximation, which is unpractical in the circumstances of our interest.

In order to solve this issue, we employ a deep normalizing flow (DNF) to construct $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$. Compared with other generative models, e.g., variational auto-encoders (VAEs) and generative adversarial networks (GANs), DNF is a fully

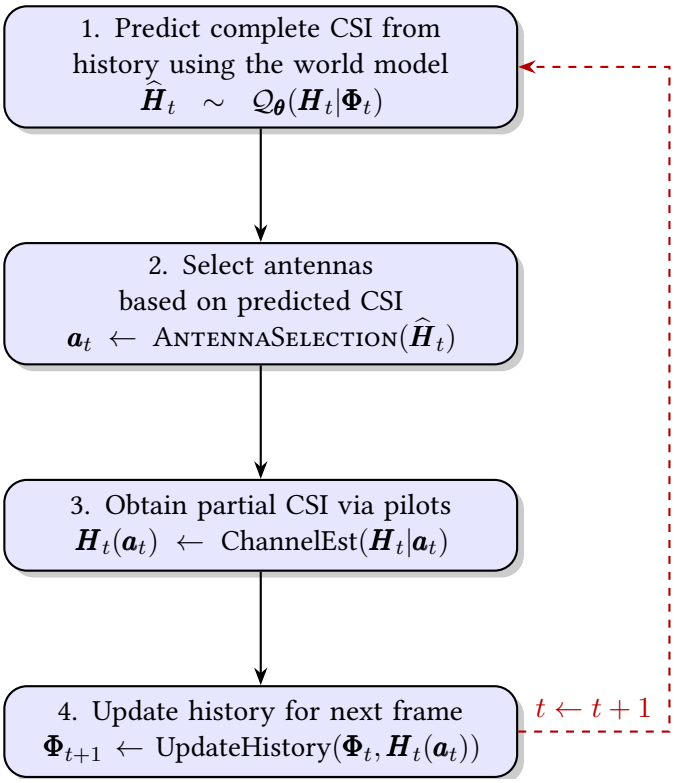

Figure 3.2 Workflow of the proposed framework.

probabilistic model with tractable exact density inference, which can accelerate the search efficiency of Monte-Carlo tree search [73]. Although in this paper the exact density of the predicted channel states is not utilized, it can help reduce the number of simulations in our future work by evaluating the certainty of current solutions. As a kind of generative model, DNF approximates the distribution by the change of latent distribution. This strategy allows us to sample complete observations from the latent space, while still being able to compute the corresponding log-likelihood by the law of change of variables [74, 75]. In principle, DNF assumes that complete observation $\boldsymbol{H}_t$ depends on a latent random variable $\boldsymbol{Z}_t$ following a tractable distribution $\mathcal{P}_{\boldsymbol{\omega}}(\boldsymbol{Z}_t)$, where $\boldsymbol{\omega}$ is the parameters of the latent distribution. It is also assumed that $\boldsymbol{\omega}$ follows a tractable distribution, denoted by $\boldsymbol{\omega} \sim \mathcal{P}_{\boldsymbol{\psi}}(\boldsymbol{\omega})$ with parameters $\boldsymbol{\psi}$. Besides, the parameters $\boldsymbol{\psi}$ can be determined based on the history, i.e., $\boldsymbol{\psi} = \gamma_{\boldsymbol{\theta}_1}(\boldsymbol{\Phi}_t)$ in which $\gamma_{\boldsymbol{\theta}_1}(\cdot)$ is a function represented by a deep

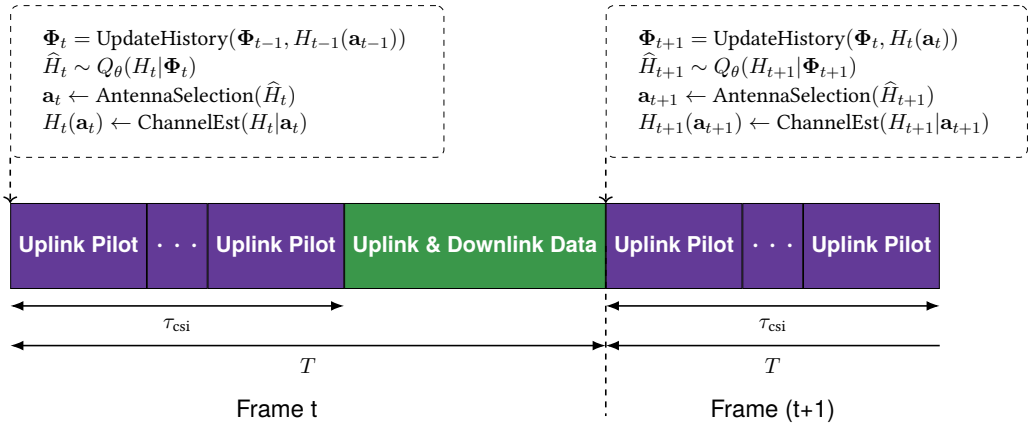

Figure 3.3 Time horizon of the proposed framework.

neural network (DNN) with parameters $\boldsymbol{\theta}_1$.

Intuitively, this approach takes the uncertainty of incomplete observations into channel prediction by considering $\boldsymbol{\omega}$ as a random variable conditioned on the given history $\mathbf{\Phi}_t$. Therefore, the latent space is also conditioned on incomplete observations, denoted as $\boldsymbol{Z}_t \sim \mathcal{P}_{\mathcal{Z}}(\boldsymbol{Z}_t|\mathbf{\Phi}_t)$. In conclusion, the generative process can be described as

$$\begin{aligned} \boldsymbol{\psi}_t &= \gamma_{\boldsymbol{\theta}_1}(\mathbf{\Phi}_t)'; & \boldsymbol{\omega}_t &\sim \mathcal{P}_{\boldsymbol{\psi}_t}(\boldsymbol{\omega}_t) \\ \boldsymbol{Z}_t &\sim \mathcal{P}_{\boldsymbol{\omega}_t}(\boldsymbol{Z}_t); & \boldsymbol{H}_t &= g_{\boldsymbol{\theta}_2}(\boldsymbol{Z}_t), \end{aligned} \tag{3.12}$$

where $g_{\boldsymbol{\theta}_2}(\cdot)$ represents an *invertible (or bijective)* function with parameters $\boldsymbol{\theta}_2$. Since $g_{\boldsymbol{\theta}_2}(\cdot)$ is an invertible function, the associated latent variable can be effectively inferred by $\boldsymbol{Z}_t = f_{\boldsymbol{\theta}_2}(\boldsymbol{H}_t) \triangleq g^{-1}_{\boldsymbol{\theta}_2}(\boldsymbol{H}_t)$. By setting $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$, the log-likelihood of the complete observation $\boldsymbol{H}_t$ can be approximately computed from

$$\log \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\mathbf{\Phi}_t) = \log \mathcal{P}_{\mathcal{Z}}(f(\boldsymbol{H}_t)|\mathbf{\Phi}_t) + \log\Big|\det\Big(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{H}_t}\Big)\Big|, \tag{3.13}$$

where $\det\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{H}}\right)$ is the determinant of the Jacobian. In order to construct a flexible

model $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$, we assume that the invertible function $f(\cdot)$ is composed by N_I invertible sub-functions, given by

$$f(\cdot) = f_1(\cdot) \otimes f_2(\cdot) \otimes \cdots f_n(\cdot) \cdots \otimes f_{N_I}(\cdot). \tag{3.14}$$

Based on the above factorization, we can infer the corresponding latent variable $\boldsymbol{Z}$ accordingly,

$$\boldsymbol{H} \xrightarrow{f_1} \boldsymbol{V}_1 \xrightarrow{f_2} \boldsymbol{V}_2 \cdots \xrightarrow{f_n} \boldsymbol{V}_n \cdots \xrightarrow{f_{N_I}} \boldsymbol{Z}. \tag{3.15}$$

Now, using the notations $\boldsymbol{V}_0 \triangleq \boldsymbol{H}_t$ and $\boldsymbol{V}_{N_i} \triangleq \boldsymbol{Z}_t$, we can rewrite (3.13) as

$$\begin{aligned} \log \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t) = &\log \mathcal{P}_{\mathcal{Z}}(f(\boldsymbol{H}_t)|\boldsymbol{\Phi}_t) \\ &+ \textstyle\sum_{n=1}^{N_I} \log \Big| \det \big(\frac{\mathrm{d}\boldsymbol{V}_n}{\mathrm{d}\boldsymbol{V}_{n-1}} \big) \Big|. \end{aligned} \tag{3.16}$$

Thus, we construct the approximate model $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ with N sub-functions, with each sub-function being a small flow step of the complete flows. In this case, it is straightforward to train the model $\mathcal{Q}_{\boldsymbol{\theta}}$ by recalling the principle of maximum likelihood approximation in (3.13).

3.3.2 Network Implementation

The applied network structure of the proposed DCGM is illustrated in Fig. 3.4. As mentioned in Sec. 3.3.1, it is important to ensure that each flow step is fully invertible for the implementation of the proposed DCGM. In our implementation, we implement the flow steps via normalization layers, invertible convolutional layers and affine coupling layers, where the details of these layers can be found in [76, 74, 77, 63]. Using the above invertible layers, an invertible network can be constructed to track channel transitions from the incomplete history. In particular, the constructed invertible network is composed by N flow steps,

and each flow step containing only three layers: activation normalization layer, invertible convolutional layer and affine coupling layer.

Because the latent variable $\boldsymbol{Z}_t$ relies on the incomplete history $\boldsymbol{\Phi}_t$, a conditional distribution $\mathcal{P}_{\mathcal{Z}}(\boldsymbol{Z}_t|\boldsymbol{\Phi}_t)$ should also be implemented. Consider $\boldsymbol{Z}_t \sim \mathcal{CN}(\boldsymbol{Z}_t|\boldsymbol{\mu}_{\boldsymbol{z}}(t), \boldsymbol{\Sigma}_{\boldsymbol{z}}(t))$ and denote $\boldsymbol{\omega}_t = \{\boldsymbol{\mu}_{\boldsymbol{z}}(t), \boldsymbol{\Sigma}_{\boldsymbol{z}}(t)\}$ as the parameters of the distribution. Then, we sample $\boldsymbol{\omega}_t$ by the reparameterization steps

$$\boldsymbol{\mu}_{\boldsymbol{z}}(t) = \boldsymbol{\mu}_{\mathbf{1}}(t) + \boldsymbol{\nu}_1 \odot \boldsymbol{\Sigma}_{\mathbf{1}}(t), \tag{3.17}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{z}}(t) = \boldsymbol{\mu}_{\mathbf{2}}(t) + \boldsymbol{\nu}_2 \odot \boldsymbol{\Sigma}_{\mathbf{2}}(t), \tag{3.18}$$

where $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are two standard complex Gaussian samples, $\odot$ denotes the element-wise multiplication, and the associated parameters

$$\boldsymbol{\psi}_t = \{\boldsymbol{\mu}_1(t), \boldsymbol{\mu}_2(t), \boldsymbol{\Sigma}_1(t), \boldsymbol{\Sigma}_2(t)\},$$

are determined by the incomplete history, i.e., $\boldsymbol{\psi}_t = \gamma_{\boldsymbol{\theta}_1}(\boldsymbol{\Phi}_t)$. Specifically, $\gamma_{\boldsymbol{\theta}_1}(\boldsymbol{\Phi}_t)$ is represented by two independent convolutional networks, which can be expressed as $\{\boldsymbol{\mu}_1(t), \boldsymbol{\Sigma}_1(t)\} = \texttt{CNN}_1(\boldsymbol{\Phi}_t)$ and $\{\boldsymbol{\mu}_2(t), \boldsymbol{\Sigma}_2(t)\} = \texttt{CNN}_2(\boldsymbol{\Phi}_t)$. $\texttt{CNN}_1(\cdot)$ and $\texttt{CNN}_2(\cdot)$ may both have exactly the same network structure composed by multiple convolutional layers and rectified linear units (ReLU) [74]. In order to retain the spatial information, zero-padding is used to keep each incomplete observation $\boldsymbol{H}_t(\boldsymbol{a}_t)$ within $\boldsymbol{\Phi}_t$ having the same shape of $N_u \times N_t$.

3.3.3 The Proposed Antenna Selection Framework

Given the prediction of complete CSI from the incomplete history, the antenna selection becomes a straightforward task. The proposed JCPAS framework utilizes the proposed channel prediction model $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ to select the antennas in practical environments. To illustrate our proposed JCPAS framework, we present its workflow and time horizon structure in Figs. 3.2 and 3.3 respectively, for the

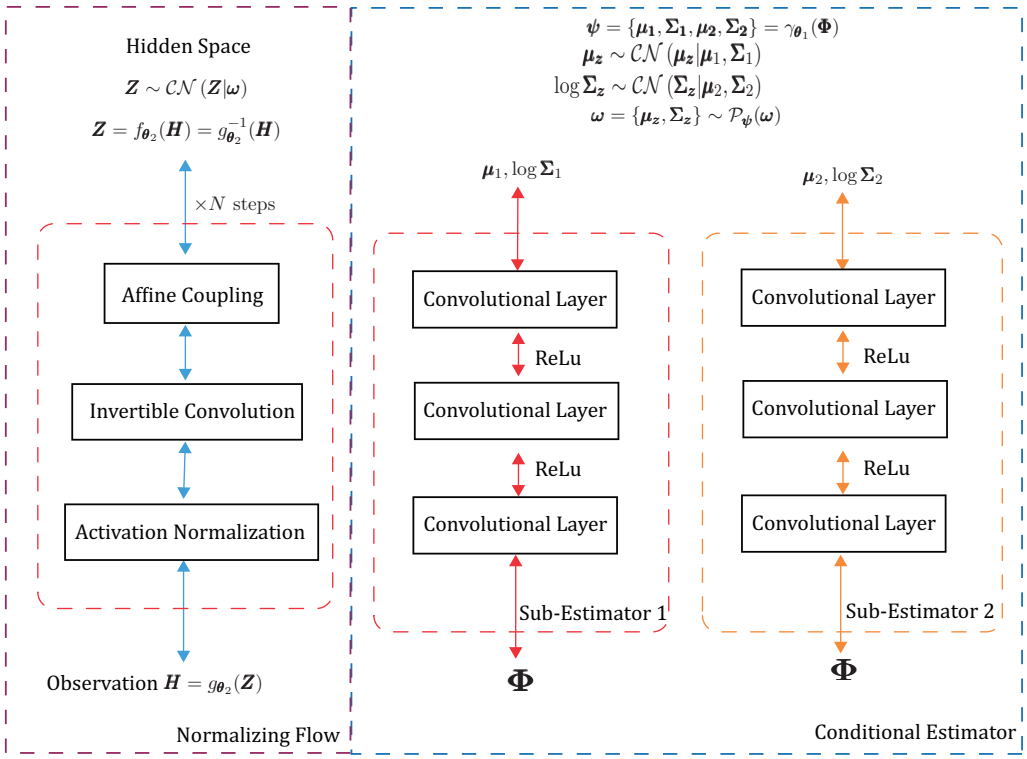

Figure 3.4 Diagram of the structure of the proposed DCGM.

ease of understanding.

At the beginning of each frame, we estimate the belief of the current channel state via the well-trained DCGM, i.e., $\widehat{\boldsymbol{H}}_t \sim \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$. It should be noted that we fill in the history by zero entries for initialization. After estimating the belief state, we employ an antenna selection algorithm (e.g., Algorithm 6) to select the antennas subset for acquiring the incomplete CSI as well as data transmission. When the data transmission of the current frame is completed, we update the history for the next frame by pushing the last incomplete observation $\boldsymbol{H}_t(\boldsymbol{a}_t)$ into $\boldsymbol{\Phi}_t$.

To summarize the process of our proposed JCPAS framework, the associated pseudo-code is detailed in Algorithm 7. It is worth noting that the proposed framework is a general framework with the purpose of reducing the channel estimation overhead for massive MIMO systems. Thus, the choice of antenna

Algorithm 7: Proposed Joint Channel Prediction and Antenna Selection (JCPAS) Framework

$t \leftarrow 0$;
Initialize history $\boldsymbol{\Phi}_t \leftarrow \{\mathbf{0}_{N_u \times N_t}, \ldots, \mathbf{0}_{N_u \times N_t}\}$;
while *true* **do**
  $\widehat{\boldsymbol{H}}_t \sim \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t | \boldsymbol{\Phi}_t)$;
  $\boldsymbol{a}_t \leftarrow$ AntennaSelection($\widehat{\boldsymbol{H}}_t$) (e.g., Algorithm 6);
  Obtain the incomplete channel measurements $\boldsymbol{H}_t(\boldsymbol{a}_t)$ with $\boldsymbol{a}_t$;
  Precoding($\boldsymbol{H}(\boldsymbol{a}_t)$);
  DataTransmission($\boldsymbol{H}_t(\boldsymbol{a}_t)$);
  for $i \leftarrow 0$ **to** $\min(t, L)$ **do**
    $\boldsymbol{\Phi}_t[L-i] \leftarrow \boldsymbol{H}_{t-i}(\boldsymbol{a}_{t-i})$;
  end
  $t \leftarrow t+1$;
end

selection algorithms is not limited, and hereby we use the greedy search algorithm for illustration. In principle, the choice of antenna selection algorithms should be determined based on the available resources and different needs of environments.

3.4 Proposed Risk-Aware Planing Antenna Selection Algorithm

This section will introduce the proposed risk-aware planning algorithm, which can be integrated into the JCPAS framework to further improve system robustness, and meanwhile reducing the chance of violating practical constraints. After that, we will discuss the performance-complexity tradeoff of the proposed algorithm.

3.4.1 Risk-Aware Monte Carlo Tree Search

Due to incomplete CSI and imperfect belief estimation $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$, antenna selection often carries the risk of violating practical constraints of problem (3.3), as the information is not perfect for making decisions. In principle, the outcome $\mathcal{F}(\boldsymbol{a}_t)$

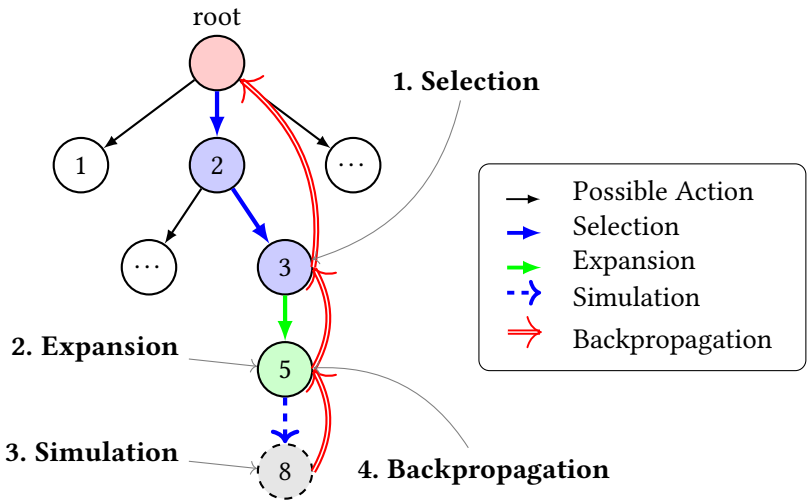

Figure 3.5 The four steps of the MCTS algorithm shown on a single tree: A path is **selected** using UCT, a new node is **expanded**, a fast and often random **simulation** (rollout) is run, and the result is **backpropagated** to update node statistics.

of the selected antennas $\boldsymbol{a}_t$ can be regarded as a random variable conditioned on the history $\boldsymbol{\Phi}_t$. This indicates that making decisions based on the inspection of a single sample from the estimated belief distribution is obviously not sufficient and risky.

To be specific, such potential risk results from two parts: the probability of selecting antennas based on a sampled belief deviating from the truth, and the negative consequences if it does. Unfortunately, the existing algorithms depend on the known channel coefficients rather than the channel statistics. To solve this issue, planning based on the expected outcomes should play a crucial role in risk-aware AS algorithm design. This realization is a key to heuristic risk-aware planning in the presence of incomplete observations and to guarantee the constraints.

From this perspective, we propose to select antennas and manage risks (of violating the system constraints) by learning a posterior distribution over the expected outcomes of each selected antennas combination. This approach can be accomplished based on the concepts of MCTS [40] and bootstrap Thompson

sampling (BTS) [27, 78, 79, 26]. As a best-first search strategy, MCTS employs a heuristic exploration to iteratively explore the combinatorial search space. In general, the search space can be regarded as a decision tree consisting of decision nodes, and each of which has a number of child nodes, and each child node corresponds to an available decision of removing the associated antenna index. As illustrated in Fig. 3.5, the typical routine of MCTS includes the following four steps [40]:

- **Selection:** We traverse the search tree in accordance to the estimated statistics of each node until encountering a node that has not been fully expanded, which is also called as a leaf node.
- **Expansion:** Whenever a leaf node is selected, it must be expanded. The expansion is done by randomly generating a child node and then initializing the prior information for the newly generated node.
- **Simulation:** We execute a random rollout through Monte Carlo simulations until a complete selection is reached. We then simulate the outcome of the explored complete selection by Monte Carlo simulation.
- **Backpropagation:** After receiving the simulated outcome of a complete selection, the results will be backpropagated to all ascendant nodes, in which a set of predefined algorithm statistics should be updated accordingly.

During the search, the above routine will be repeated a number of times such that the combinatorial search space will be incrementally explored and the search tree will be simultaneously expanded. After a number of simulations, the statistics of each node will be sufficient for making decisions.

As described in Algorithm 8-10, we jointly use $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ and MCTS to build the proposed RA-MCTS algorithm. The proposed RA-MCTS works in a similar manner as the greedy search, which removes antennas one by one iteratively. Each rollout is simulated based on the sample drawn from the estimated belief

distribution $\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$. Besides, we maintain an approximate posterior distribution over the expected outcome of simulations, and utilize BTS to compute policies in risk-aware and multi-constraints settings.

For each simulation, the proposed RA-MCTS always starts at a root node with a belief sample $\widehat{\boldsymbol{H}}_t \sim \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t|\boldsymbol{\Phi}_t)$ drawn from the estimated belief distribution. It is worth nothing that the belief sample $\widehat{\boldsymbol{H}}_t$ is utilized to compute the prior probability of each node. Specifically, we start with the antenna set $\boldsymbol{a} = \{1, 2, \ldots, N_t\}$ and define $\boldsymbol{A} = (\widehat{\boldsymbol{H}}_t\widehat{\boldsymbol{H}}_t^H)^{-1}$. Note that we will sometimes omit the time index t since the selection is done within the current frame. Then, we update the antenna set as $\boldsymbol{a} = \boldsymbol{a} \setminus \{m\}$ whenever an antenna index m is removed. Similar to greedy search, $\mathbf{A}$ can also be successively updated as $\boldsymbol{A} = \boldsymbol{A} + \frac{\boldsymbol{A}\widehat{\boldsymbol{h}}_m\widehat{\boldsymbol{h}}_m^H\boldsymbol{A}}{1-\widehat{\boldsymbol{h}}_m^H\boldsymbol{A}\widehat{\boldsymbol{h}}_m}$. When expanding a new child node i, we compute the corresponding potential spectral loss as $\lambda_i = \frac{\left\|\boldsymbol{h}_i^H\boldsymbol{A}\right\|^2}{1-\boldsymbol{h}_i^H\boldsymbol{A}\boldsymbol{h}_i}$. In addition, we have $\alpha_i = \frac{e^{-\lambda_i}}{\sum_{j\neq i} e^{-\lambda_j}}$, which denotes the prior probability of exploring child node i.

By considering each selection as a multi-arm bandit problem, we use the concept of BTS to select child nodes while balancing between exploration and exploitation. The intuition behind BTS is simple and intuitive. The algorithm randomly selects a child node at each step with the probability of being optimal according to current beliefs and, in the meantime, continues to sample all possible child nodes that could plausibly be optimal [79, 27]. As more information is collected, beliefs about the expected utility of each node are carefully tracked to balance exploration and exploitation [26]. In contrast to the existing Thompson sampling based AS scheme introduced in [31], our proposed approach does not rely on the assumption of equal antenna contribution, as well as the Beta-Bernoulli posterior specification. Indeed, these advantages can help improve the robustness under the circumstance of model misspecification.

Specifically, we adopt a bootstrap distribution to approximate the posterior distribution over the expected outcome of each node. The bootstrap distribution is parameterized by a number of bootstrap replicates, $j \in \{1, \ldots, J\}$. In the

Algorithm 8: Risk-Aware Monte Carlo Tree Search (RA-MCTS)

Input: History of the past incomplete measurements $\boldsymbol{\Phi}_t$
Output: Selected antenna combination $\boldsymbol{a}_t$

Let the set of all antennas be $\boldsymbol{\mathcal{N}} = \{1, \ldots, N_t\}$ and set of removed antennas as $\boldsymbol{r} = \emptyset$;
while $|\boldsymbol{r}| < N_t - N_f$ **do**
  $m \leftarrow \text{RiskAwarePlanning}(\boldsymbol{r}, \boldsymbol{\Phi}_t)$;
  $\boldsymbol{r} \leftarrow \boldsymbol{r} \cup \{m\}$;
end
$\boldsymbol{a}_t \leftarrow \{1, 2, \ldots, N_t\} - \boldsymbol{r}$;
return $\boldsymbol{a}_t$;
Function *RiskAwarePlanning($\boldsymbol{r}$, $\boldsymbol{\Phi}_t$)*
  while *within computational budget* **do**
    $\widehat{\boldsymbol{H}}_t \sim \mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{H}_t | \boldsymbol{\Phi}_t)$;
    $\boldsymbol{A} \leftarrow (\widehat{\boldsymbol{H}}_t \widehat{\boldsymbol{H}}_t^H)^{-1}$;
    $n \leftarrow \text{RetrieveNode}(\boldsymbol{r}, \emptyset, \boldsymbol{A}, \widehat{\boldsymbol{H}}_t)$;
    $v = \text{TreePolicy}(n)$;
    $\delta = \text{Rollout}(v)$;
    $\text{Backpropagation}(\delta, v)$;
  end
  return *BestChild*(n);
Function *RetrieveNode($\boldsymbol{r}, \boldsymbol{r}', \boldsymbol{A}, \widehat{\boldsymbol{H}}$)*
  if *the node corresponding to $\boldsymbol{r}$ has not been generated* **then**
    for $m \in \boldsymbol{r} - \boldsymbol{r}'$ **do**
      $\boldsymbol{A} \leftarrow \boldsymbol{A} + \frac{\boldsymbol{A}\widehat{\boldsymbol{h}}_m \widehat{\boldsymbol{h}}_m^H \boldsymbol{A}}{1 - \widehat{\boldsymbol{h}}_m^H \boldsymbol{A} \widehat{\boldsymbol{h}}_m}$;
    end
    if $|\boldsymbol{r}| < N_t - N_f$ **then**
      $\lambda_i \leftarrow \frac{\left\| \boldsymbol{h}_i^H \boldsymbol{A} \right\|^2}{1 - \boldsymbol{h}_i^H \boldsymbol{A} \boldsymbol{h}_i}, \ \ \forall i \in \boldsymbol{\mathcal{N}} - \boldsymbol{r}$;
      $\alpha_i \leftarrow \frac{e^{-\lambda_i}}{\sum_{j \neq i} e^{-\lambda_j}}, \ \ \forall i \in \boldsymbol{\mathcal{N}} - \boldsymbol{r}$;
      for *each child node* $i \in \boldsymbol{\mathcal{N}} - \boldsymbol{r}$ **do**
        $\alpha_{ij} \leftarrow \alpha_i, \ \ \forall j \in \{1, \ldots, J\}$;
        $\beta_{ij} \leftarrow 1, \ \ \forall j \in \{1, \ldots, J\}$;
      end
      return *a node with the above associated statistics*;
  return *the already generated node*;

Algorithm 9: Utility Functions of RA-MCTS

Function *TreePolicy*(n)
  while *node* n *is not a terminal node* **do**
    if *node* n *is not fully expanded* **then**
      return *Expand*(n);
    else
      $n \leftarrow \text{BestChild}(n)$;
    end
  end
  return n;
Function *Expand*(n)
  $\boldsymbol{r}, \boldsymbol{A}, \widehat{\boldsymbol{H}} \leftarrow n$;
  $\mathcal{C} \leftarrow \emptyset$;
  for *each untried child node* $i \in \boldsymbol{\mathcal{N}} - \boldsymbol{r}$ **do**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$;
  end
  $m \leftarrow \arg\max_{i \in \mathcal{C}} \alpha_i$;
  return *RetrieveNode*($\boldsymbol{r} \cup \{m\}, \boldsymbol{r}, \boldsymbol{A}, \widehat{\boldsymbol{H}}$);
Function *BestChild*(n)
  $\boldsymbol{r}, \boldsymbol{A}, \widehat{\boldsymbol{H}} \leftarrow n$;
  for *each child node* $i \in \boldsymbol{\mathcal{N}} - \boldsymbol{r}$ **do**
    Sample uniform replicate $j \in \{1, 2, \ldots, J\}$;
    Retrieve α_{ij}, β_{ij} according to j;
  end
  $m \leftarrow \arg\max_i \frac{\alpha_{ij}}{\beta_{ij}}$; ; `// The index` j `is the one sampled in the last iteration of the loop above`
  return *RetrieveNode*($\boldsymbol{r} \cup \{m\}, \boldsymbol{r}, \boldsymbol{A}, \widehat{\boldsymbol{H}}$);

initialization of a new node i, for each bootstrap replicate, j, we store a set of parameters with $\alpha_{ij} = \alpha_i$ and $\beta_{ij} = 1$ by default, and these parameters will be updated during the backpropagation of each simulation. Specifically, at node i, for each bootstrap replicate j, we update α_{ij} and β_{ij} depending on the result of a coin flip $\texttt{Bernoulli}(\frac{1}{2})$. If a coin flip is equal to 1, we update the parameters by

$$\alpha_{ij} = \alpha_{ij} + \mathcal{U}(\boldsymbol{a}); \quad \beta_{ij} = \beta_{ij} + 1, \tag{3.19}$$

Algorithm 10: Utility Functions of RA-MCTS

Function *Rollout*(n)
  while *node* n *is not a terminal node* **do**
    $n \leftarrow \text{BestChild}(n)$;
  end
  $\boldsymbol{r}, \boldsymbol{A}, \widehat{\boldsymbol{H}} \leftarrow n$;
  $\boldsymbol{a} \leftarrow \boldsymbol{\mathcal{N}} - \boldsymbol{r}$;
  return $\mathcal{U}(\boldsymbol{a})$;
Function *Backpropagation*(δ, n)
  while *node* n *is not null* **do**
    UpdateDistribution(δ, n);
    $n \leftarrow$ parent of n;
  end
Function *UpdateDistribution*(δ, i)
  for $j \in \{1, 2, \ldots, J\}$ **do**
    Sample $\epsilon_j \sim \texttt{Bernoulli}(\frac{1}{2})$;
    if $\epsilon_j = 1$ **then**
      $\alpha_{ij} \leftarrow \alpha_{ij} + \delta$;
      $\beta_{ij} \leftarrow \beta_{ij} + 1$;
    end
  end

where $\mathcal{U}(\boldsymbol{a})$ denotes a utility function which evaluates the normalized outcome of a complete antenna selection $\boldsymbol{a}$ (i.e., $|\boldsymbol{a}| = N_f$) in the simulation phase. For instance, a utility function in the energy minimization problem is given by

$$\mathcal{U}(\boldsymbol{a}) = \begin{cases} 1 - \frac{\sum_{k=1}^{N_u} \|\boldsymbol{w}_k(\boldsymbol{a})\|^2}{P_{tot}} & \text{if constraints satisfied,} \\ 0, & \text{else,} \end{cases} \tag{3.20}$$

which normalizes the risk of violating the QoS requirements in the outcome evaluation. This implies that if it cannot satisfy all the constraints, the BS will try its best to allocate all the available power budgets to improve QoS. To decide which antenna should be removed, the previously computed statistics is utilized. For each child node i, we first uniformly sample j from J bootstrap replicates,

and then a node with the largest point estimate is selected by $i^* = \arg\max_i \frac{\alpha_{ij}}{\beta_{ij}}$, which allows us to break ties randomly.

3.4.2 Complexity Analysis

Intuitively, our proposed RA-MCTS degrades to greedy search (see Algorithm 6) if the computational budget of each iteration is restricted to one single simulation. On the other hand, RA-MCTS is capable of intelligently managing risks and allocating exploration efforts with sufficient computational budget, while the greedy search ignores risks and does not actively explore. This implies that our proposed RA-MCTS achieves a trade-off between performance and computational complexity.

In general, the computational complexity of our proposed RA-MCTS depends on two aspects: the number of simulations and the computational cost of each simulation. For the latter, it can be controlled by the number of bootstrap replicates, as the choice of J obviously limits the number of samples we have from the bootstrap distribution. For a smaller J, RA-MCTS is expected to become greedy. This is because the probability of choosing some nodes that do not have the largest point estimate will show a trend of being zero. On the other hand, a larger J will involve more exploration, at an expense of extra computation complexity.

To characterize the computational complexity of our proposed algorithms, we consider the total number of floating-point operations (a.k.a. FLOPs) to quantify the complexity order. Note that we consider a FLOP to be either a complex-valued multiplication or a complex-valued summation. In fact, a complex-valued multiplication requires 4 real-valued multiplications and 2 real-valued summations, whereas a complex-valued summation requires only 2 real summations. However, each operation will be counted as one FLOP for simplicity. Given $\boldsymbol{A} \in \mathbb{C}^{q\times p}$ and $\boldsymbol{X} \in \mathbb{C}^{p\times r}$, the arithmetic order of FLOPs for the matrix mutiplication of $\boldsymbol{AX} \in \mathbb{C}^{q\times r}$ is of $\mathcal{O}(pqr)$. Given a nonsingular $\boldsymbol{Y} \in \mathbb{C}^{n\times n}$, the computation

Name of Methods	Computational Complexity
Online [31]	$\mathcal{O}(N_t \log N_t + N_f)$
Greedy Search [56]	$\mathcal{O}(N_u^2(N_t^2 - N_f^2))$
JCPAS-Basic	$\mathcal{O}(N_u^2(N_t^2 - N_f^2) + P_{Net})$
RA-MCTS(N, J)	$\mathcal{O}(NN_u^2(N_t^2 - N_f^2) + NP_{Net} + NN_t(N_t - N_f)J)$
Exhaustive Search[56]	$\mathcal{O}\Big(N_u^2 N_t + \frac{N_t^{N_f}}{N_f!}(N_t - N_f) + 2N_u(N_t - N_f)^2 + (N_t - N_f)^3\Big)$

Table 3.1 Computational Complexity Comparisons of Antenna Selection Algorithms with Zero-Forcing Precoding.

complexity for the matrix inversion $\boldsymbol{Y}^{-1}$ is of $\mathcal{O}(n^3)$.

For the RA-MCTS algorithm, it computes and stores the result of $(\boldsymbol{H}\boldsymbol{H}^H)^{-1}$ only once, which contains one matrix multiplication and one matrix inversion. The complexity order of $(\boldsymbol{H}\boldsymbol{H}^H)^{-1}$ is $\mathcal{O}(N_u^2 N_t)$. For each loop of RA-MCTS, despite the fact that it works in a similar manner as greedy search, however, it should be noted that RA-MCTS may explore a path which starts from an intermediate node and has been partially explored before, while greedy search always explore a brand new path starting from the root. This difference implies that for each loop of RA-MCTS, the complexity of greedy search can be regarded as an upper bound in the worst cases, which is given by $\mathcal{O}(N_u^2(N_t^2 - N_f^2))$ [56]. Since RA-MCTS has to backpropagate the simulation results up to $N_t - N_f$ ascendant nodes, the computational complexity for the tree policy and rollout procedures is bounded by $\mathcal{O}\left(N_t(N_t - N_f)J + N_u^2(N_t^2 - N_f^2)\right)$.

Besides, RA-MCTS also draws one belief state from the belief distribution at each loop. As to the neural network, the normalizing flow is constructed by three kinds of invertible layers, and their computational complexities are determined by element-wise operations and log-determinants [76, 74, 77]. Hence, for a normalizing flow with N_I layers, the computational complexity depends on the input size, which is given by $\mathcal{O}((N_I L N_t N_u)$. For a CNN with L_{Conv} layers, we denote the kernel size and the number of kernels at the i-th layer as $S_{ker}(i)$ and $N_{ker}(i)$, respectively. Then, the computational complexity of CNN is given

Band	Scenarios	Speed (km/h)	Bandwidth (kHz)	Frame duration (ms)	Correlation coefficient
LTE@2.6 GHz	Urban	27.0	15	~ 15.3	0.990
LTE@2.6 GHz	Urban	36.0	15	~ 11.5	0.986
LTE@2.6 GHz	Highway	140.0	15	~ 2.2	0.950
LTE@2.6 GHz	Railway	290.0	15	~ 1.4	0.900

Table 3.2 Temporal Correlation Coefficients of Typical Scenarios [19].

by $\mathcal{O}(\sum_{i=1}^{L_{Conv}} N_{ker}(i-1)S_{ker}(i)^2 N_t N_u N_{ker}(i))$ [80]. The total computational complexity of drawing belief samples is given by:

$$P_{Net} = \mathcal{O}\Big(N_I L N_t N_u + \sum_{i=1}^{L_{Conv}} N_{ker}(i-1)S_{ker}(i)^2 N_t N_u N_{ker}(i)\Big). \tag{3.21}$$

Consider RA-MCTS(N, J) with N rollouts and J replicas in total, its computational complexity can be bounded by:

$$\mathcal{O}\left(N P_{Net} + N N_t (N_t - N_f) J + N N_u^2 (N_t^2 - N_f^2)\right). \tag{3.22}$$

For comparison, we summarize the computational complexities of some antenna selection algorithms in Table 3.1. It is clear that JCPAS-Basic is a special case of RA-MCTS(N, J) with $N = 1$ and $J = 1$. Moreover, JCPAS-Basic works in the same way as greedy search, except that the complete CSI is predicted from the belief distribution.

3.5 Numerical Results

3.5.1 Environment Setup

We perform simulations considering the energy minimization problem in (3.3), and the utility function (3.20) is adopted to evaluate the system energy efficiency subjected to limited transmit power and QoS constraints. The users are assumed

to be randomly located around the BS, and the channels between the BS and users are time-varying. In order to simulate the real propagation environment, we consider temporally correlated fading channels. By the maximum entropy principle, it is common to characterize the channel evolution by the Gaussian-Markov process with a one-step correlation coefficient given by Jakes' model [65, 66, 67]. Mathematically, the first-order Gaussian-Markov channel model is described by $\boldsymbol{h}_{k,t} = \zeta_k \boldsymbol{h}_{k,t-1} + \sqrt{1-\zeta_k^2}\boldsymbol{\Delta}_t$, where $\zeta_k \in [0,1]$ denotes the temporal correlation coefficient for user k, and $\boldsymbol{\Delta}_t$ is the innovation process with unit-variance complex Gaussian i.i.d. in time. The value of ζ_k is determined by the maximum Doppler frequency and is inversely proportional to the terminal speed [81], in which $\zeta_k = 1$ represents a static channel and $\zeta_k = 0$ implies that the channel is i.i.d. over time. The fading correlation coefficient can be obtained from Jakes' model given by $\zeta_k = J_0\left(2\pi \frac{v_k f_c}{C} T\right)$, where $J_0(\cdot)$ denotes the zeroth order Bessel function of the first kind, v_k is the speed of user k, C is the speed of light, and T is the frame duration [81]. For example, we present the correlation coefficients of some typical scenarios within a wide range of speeds from 3.6 km/h to 290 km/h in Table 3.2. Note that we consider the following two scenarios in our simulations:

- **Scenario I**, low-mobility users: Each user has a uniformly distributed random fading correlation coefficient as $\zeta_k \sim \texttt{Uni}(0.997, 0.999)$, which represents pedestrians and runners.

- **Scenario II**, high-mobility users: Each user has a uniformly distributed random fading correlation coefficient as $\zeta_k \sim \texttt{Uni}(0.932, 0.982)$, which represents the vehicular terminals.

In terms of the structure of the neural network, we employ a normalizing flow with $N_i = 16$ flow steps, and each CNN contains 6 layers where the number of convolutional kernels and the kernel size of each layer are $\{64, 32, 32, 16, 64, 128\}$ and $\{3, 9, 3, 3, 3, 9\}$, respectively. Note that the parameters of the neural network

are chosen on the basis of experimental experiences. "The DNF model is trained based on the principle of maximum likelihood approximation given in (14) using Pytorch machine learning framework and Adam optimizer." Other common parameters are as follows: $N_f = 8, N_u = 6, P_{tot} = 20$ dBW, $L = 128$.

3.5.2 Competing Algorithms

In order to verify the effectiveness of the proposed framework, we compare the proposed algorithms with various competitive algorithms. The proposed algorithms are summarized below:

- **JCPAS-Basic**: The proposed JCPAS framework introduced in Section 3.3.1, where the greedy search algorithm is used to select antennas.
- **RA-MCTS(**N**,** J**)**: The proposed RA-MCTS algorithm introduced in Section 3.4.1, where N is the maximum number of simulations and J is the number of bootstrap replicates. When compared to JCPAS-Basic, the only difference is that the greedy search algorithm is replaced by RA-MCTS in the JCPAS framework.

We compare our proposed algorithms with three following benchmark schemes:

- **Random**: This scheme randomly selects antennas for data transmission, which is the most naive solution.
- **Online [31]**: The online antenna selection algorithm uses Thompson sampling to select antennas with incomplete CSI.
- **Full CSI**: This scheme uses maximum channel estimation overhead to obtain the complete CSI at each frame and then employs the greedy search [56] for selecting antennas.

For a fair comparison, zero-forcing based precoding is employed in all schemes. We do not compare with the hybrid beamforming (HB) technique since it also

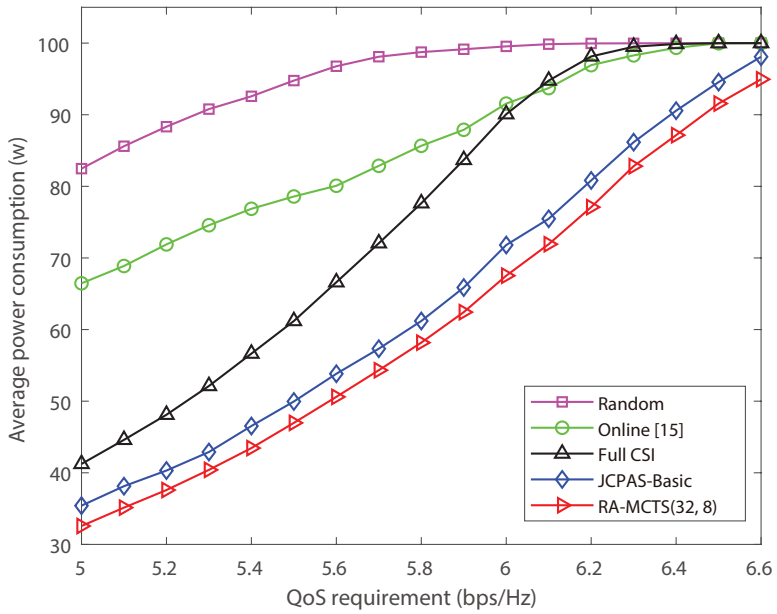

Figure 3.6 SCENARIO I, power consumption versus QoS requirements, where $N_t = 32$, $N_f = 8$, $N_u = 6$ and $T = 256$ c.u..

requires complete CSI as Full CSI reference and two schemes achieve comparable performance [82]. It is worth noting that the *Full CSI* scheme spends $\tau_{csi} = N_u\lceil\frac{N_t}{N_f}\rceil$ c.u. for CSI estimation, while other schemes use $\tau_{csi} = N_u$ c.u..

3.5.3 Performance Comparison and Discussions

Fig. 3.6 plots the average power consumption of different schemes versus the QoS requirements with $N_t = 32$, $N_f = 8$, $N_u = 6$ and $T = 256$ c.u.. It is observed from Fig. 3.6 that both the proposed JCPAS-Basic and RA-MCTS algorithms outperform the competing solutions in a wide range of QoS requirements from 5 to 6.6 bps/Hz. Specifically, when $\eta_k = 6.2$ bps/Hz, $\forall k$, the JCPAS-Basic algorithm can reduce the power consumption by about 77% and 79% compared with the "Full CSI" and "Online" schemes, respectively; while the RA-MCTS algorithm can reduce these numbers to approximately 80% and 83%. This is because the RA-MCTS selects the antennas based on the estimated posterior over the expected outcome of each antennas combination, while the JCPAS-Basic selects antennas only based on a

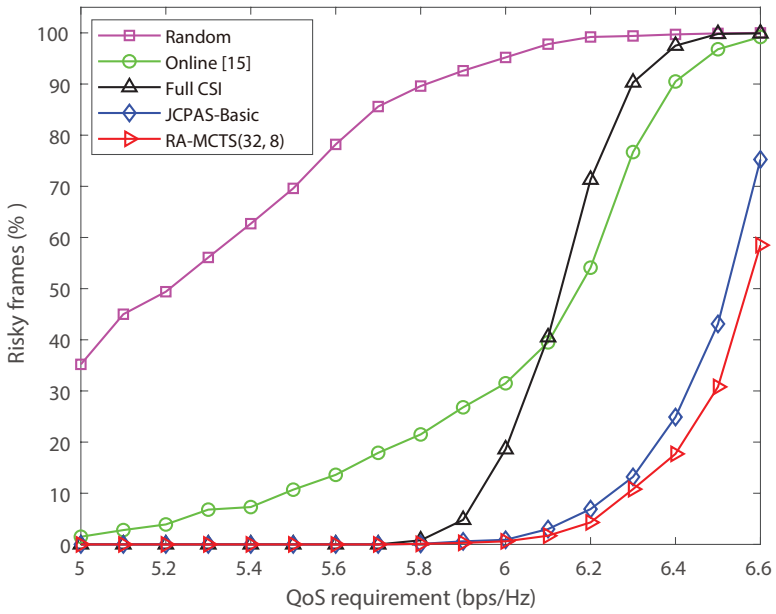

Figure 3.7 Scenario I, percentage of risky frames versus QoS requirements, where $N_t = 32$, $N_f = 8$, $N_u = 6$ and $T = 256$ c.u..

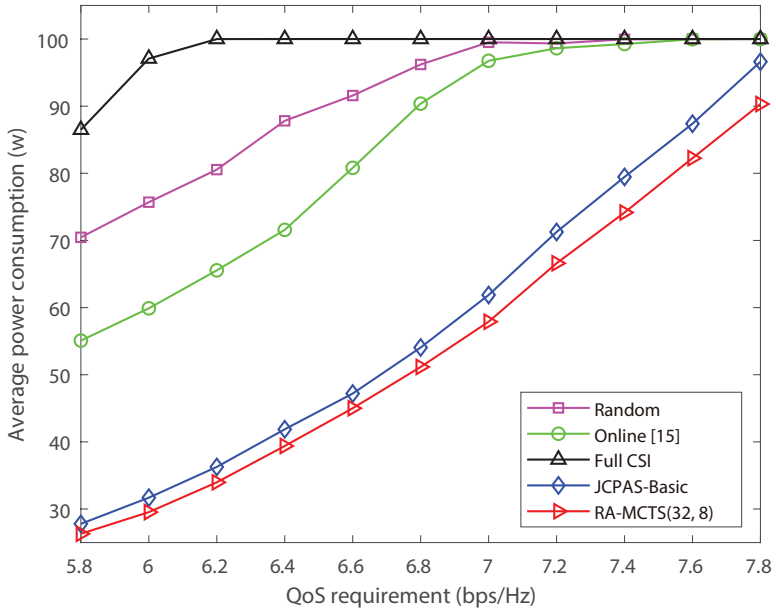

Figure 3.8 Scenario II, power consumption versus QoS requirements, where $N_t = 64$, $N_f = 8$, $N_u = 4$ and $T = 128$ c.u..

single sample from the estimated belief distribution. This difference makes the proposed RA-MCTS more robust under incomplete CSI measurements.

To demonstrate the robustness of the proposed framework, we compare the proposed algorithms with the reference schemes using a new performance metric of the percentage of risky frames in Fig. 3.7, where the simulation settings are same as in Fig. 3.6. A frame is considered as risky if any users' QoS requirement is not satisfied. It is clearly shown that the proposed algorithms, JCPAS-Basic and RA-MCTS, significantly reduce the number of risky frames compared with the references. At the spectral efficiency of 6.1 bps/Hz, all three reference schemes have more than 40% of the frames that are risky, while the proposed JCPAS-Basic and RA-MCTS algorithms have only 4% and 2% risky frames, respectively. One interesting observation is that the "Full CSI" scheme performs well at medium QoS values η_k, but its performance quickly drops for higher η_k, to be even worse than the "Online" scheme. This is because the "Full CSI" solution spends a large number of c.u. for channel estimation, hence has the least time for data transmission. As the QoS increases, the "Full CSI" could not satisfy the hight QoS requirements in the limited data transmission time even using the maximum transmit power, which results in high number of risky frames. Whereas, the "Online" scheme does not need to estimate the complete channel states. Even when the transmit power of RA-MCTS approaches the power budget (as shown in Fig. 3.6 for high η_k), the resulting percentage of risky frames is still much lower than other schemes, which demonstrates the robustness of the proposed risk-ware planning framework.

In order to further verify the effectiveness of our proposed algorithms, we present simulation results for Scenario II in Figs. 3.8 and 3.9 with $N_t = 64$, $N_f = 8$, $N_u = 4$ and $T = 128$ c.u.. Note that we use the same neural network as in Fig. 3.6. It can be observed from Fig. 3.9 that in a higher mobility scenario (reduced the frame duration) and with a larger antenna array, the average energy consumption of "Full CSI" scheme is even more than "Random", which indicates that estimating complete CSI in this case is cost-prohibitive. In general, the superior performance of the proposed algorithms are preserved compared with the references. Specifically, at $\eta_k = 6.4$ bps/Hz, the RA-MCTS scheme can reduce

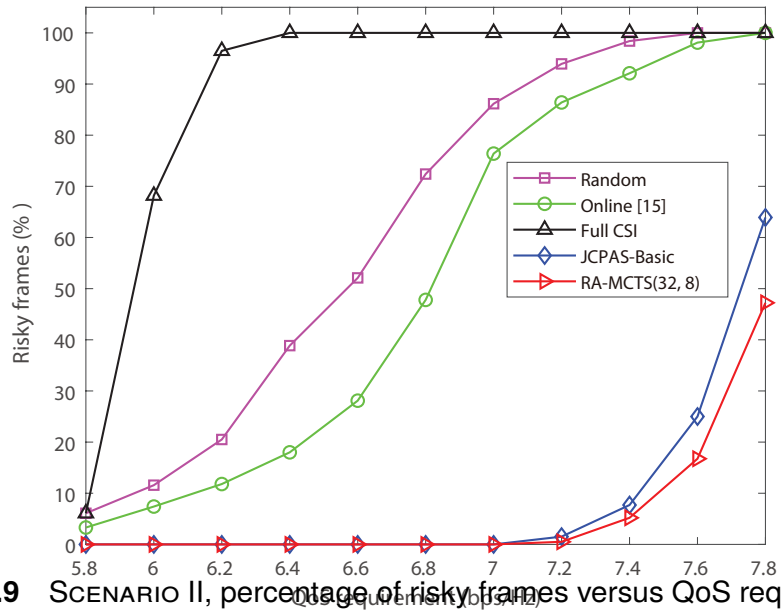

Figure 3.9 SCENARIO II, percentage of risky frames versus QoS requirements, where $N_t = 64$, $N_f = 8$, $N_u = 4$ and $T = 128$ c.u..

about 60%, 45% and 6% energy consumption compared to the "Full CSI", "Online" and JCPAS-Basic counterparts. In addition, with a very high QoS requirement of $\eta_k = 7.6$ bps/Hz, the RA-MCTS can reduce 85%, 80% and 8% risky frames compared to "Full CSI", "Online" and JCPAS-Basic. These results further verify the robustness and effectiveness of our proposed RA-MCTS.

In Figs. 3.10 and 3.11, we respectively evaluate the power consumption and the percentage of risky frames as a function of the number of users N_u, where $N_t = 128$, $N_f = 16$, $\eta_k = 6.5$ bpz/HZ and the coherence time $T = 512$ c.u.. In general, serving more users requires more transmit power at all schemes, however, the proposed JCPAS-Basic and RA-MCTS algorithms only consume about 50% of the transmit power of other schemes in most cases, as shown in Figs. 3.10. The robustness of the proposed framework is clearly shown in Figs. 3.11, in which the proposed RA-MCTS do not have any risky frame for $N_u \leq 10$, while the percentage of risky frames of three reference schemes grows quickly up to about 80% when N_u varies from 6 to 10. When $N_u = 12$, all three reference schemes have all the frames risky, while the proposed JCPAS-Basic and RA-MCTS

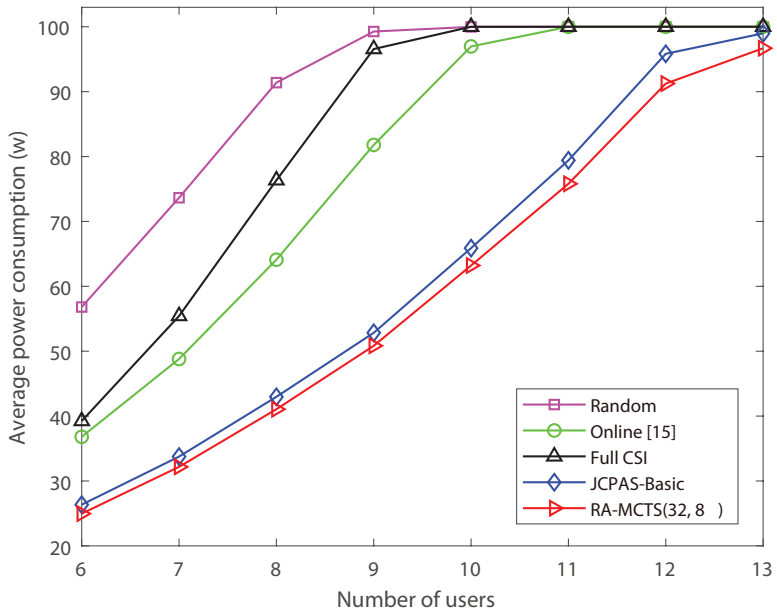

Figure 3.10 SCENARIO I, power consumption versus number of users, where $N_t = 128$, $N_f = 16$, $\eta_k = 6.5$ bps/Hz and $T = 512$ c.u..

algorithms impose a percentage of 45% and 25% of risky frames, respectively. This result confirms the robustness of the proposed RA-MCTS in highly-loaded systems with limited resources.

In Figs. 3.12-3.13, we present simulation results regarding the sum-throughput maximization optimization problem, where $N_t = 64$, $N_f = 8$, $N_u = 4$ and the user's velocity varies from 3.6 km/h to 72 km/h. In addition, the coherent time $T = 128$ c.u. and each user has a QoS requirement of 5.5 bps/Hz. Noted that if there is no feasible solution to satisfy all the constraints when optimizing the system's sum-throughput, the frame will be marked as "risky frame" and the QoS constraint will be neglected. In other words, we maximize the sum-throughput by neglecting the QoS constraints for risky frames in the simulations. From the two figures, it can be observed that the proposed algorithms outperform the other three reference schemes in terms of not only the sum-throughput but also the chance of violating user's QoS requirement. Specifically, when the power budget grows to 30 watt, JCPAS acheives 3.73 bps/Hz more than the "Online" scheme

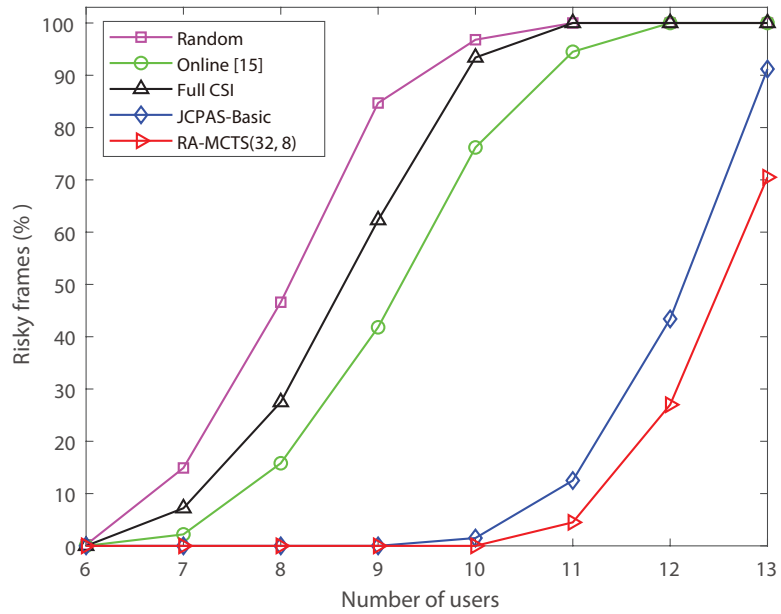

Figure 3.11 SCENARIO I, percentage of risky frames versus number of users, where $N_t = 128$, $N_f = 16$, $\eta_k = 6.5$ bps/Hz and $T = 512$ c.u..

proposed in [15], while the corresponding sum-throughput obtained by RA-MCTS is about 4.29 bps/Hz. Meanwhile, "Online" scheme has 77.10% of risky frames, while JCPAS and RA-MCTS have about only 18.40% and 5.38% risky frames. On the contrary, the proposed JCPAS and RA-MCTS are much more robust to the scenarios of our interests, as the two schemes have much higher sum-throughputs and impose much lower chance to violate the user's QoS constraint. These results show that the proposed algorithms are robust for both the power minimization problem and the sum-throughput maximization problem.

3.6 Conclusion

In this chapter, we have proposed a robust joint channel prediction and antenna selection framework JCPAS for massive MIMO systems under practical incomplete CSI, which results from the limited number of RF chains, and limited transmit power and QoS requirements. In order to address this problem, we first proposed

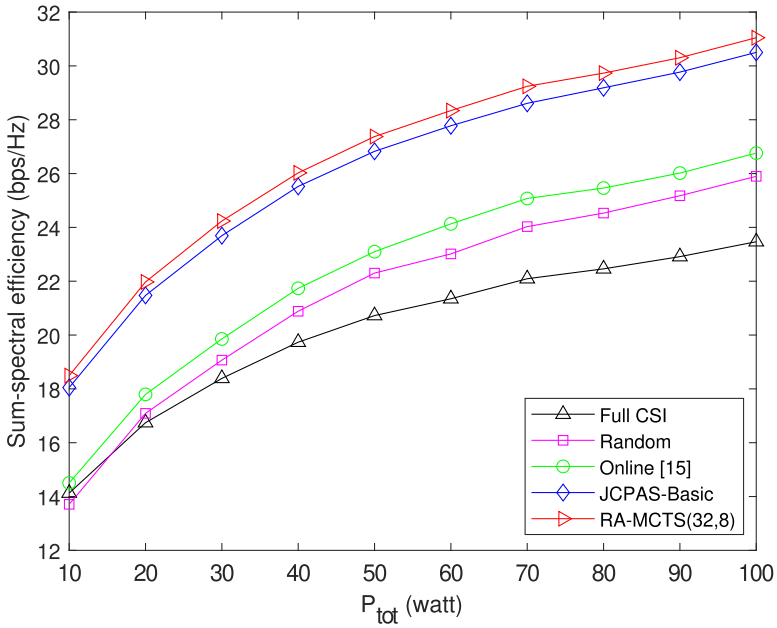

Figure 3.12 Sum-spectral efficiency versus power budget, where $N_t = 64$, $N_f = 8$, $N_u = 4$, $\eta_k = 5.5$ bps/Hz and $T = 128$ c.u..

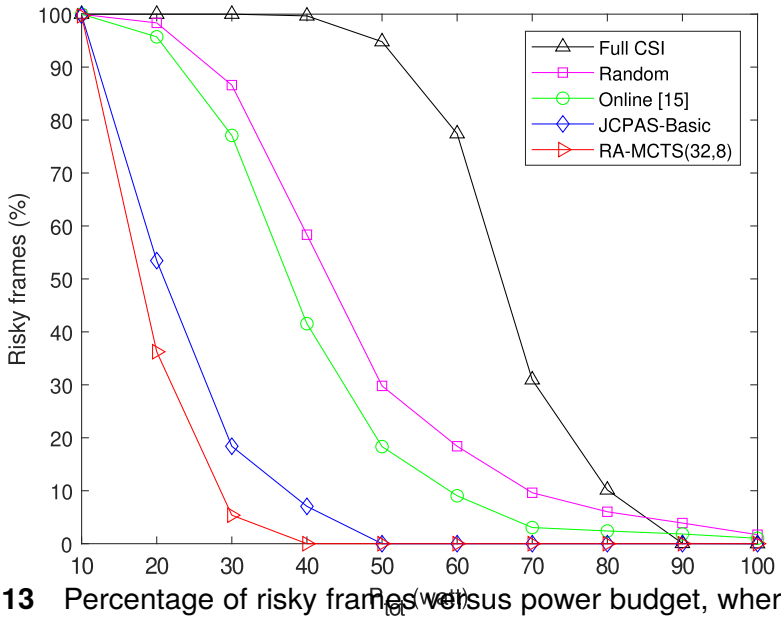

Figure 3.13 Percentage of risky frames versus power budget, where $N_t = 64$, $N_f = 8$, $N_u = 4$, $\eta_k = 5.5$ bps/Hz and $T = 128$ c.u..

a deep neural network to estimate the posterior belief distribution of the current channel states. After that, we developed a joint channel prediction and antenna

selection algorithm to select the best antennas based on the estimated belief. To improve system robustness, we proposed a risk-aware planning framework, namely RA-MCTS, which employs Monte Carlo tree search and bootstrap Thompson sampling to approximate a posterior distribution over the random objectives. Simulation results showed that the proposed RA-MCTS not only achieves a lower energy consumption but also significantly reduces the risk, quantified as the probability of violating the system constraints.

For the future work, one interesting topic is to further improve the computational efficiency of the proposed RA-MCTS. A feasible solution is to directly estimate the rollout results through a DNN such that the rollout overhead can be significantly reduced. In this case, the posterior distribution over the rollout results is also estimated by DNN, rather than the bootstrap distribution parameterized by the J replicates. The developed framework can be easily applied to cell-free MIMO systems to optimize the number of active access points under limited system resources. Another promising topic is to consider the confidence of the predicted channel coefficient in the selection process. In this case, the best antennas subset should be selected based on not only the channel gain but also the prediction confidence.

4 Build A Better World with Advanced Spatio-Temporal Predictive Learning

Prologue. In the preceding chapter, we established the effectiveness of the model-based paradigm, where the performance of the model-based risk-aware planner is fundamentally dependent on the quality of its predictive *"world model"*. An accurate prediction of the system's state from incomplete observations is crucial for effective risk-aware planning. Motivated by this, this chapter delves deeper into this critical component with the goal of "building a better world": *developing a more powerful and accurate spatio-temporal predictive model.*

We address the limitations of conventional deep learning models in capturing the complex coupled spatio-temporal dynamics found in communication system data such as MIMO channels and cellular traffic. This chapter introduces a novel architectural component, the *Crossover Attention (XOA)* mechanism, which enhances the standard Transformer model by explicitly and simultaneously processing both spatial and temporal correlations, without relying on convolutional networks. This work, based on the publication in [38], directly strengthens the model-based framework of Part I. By creating a more accurate world model, we enable the planning agent to make more informed and robust decisions, further advancing the goal of risk-aware autonomous intelligence under partial observability.

4.1 Introduction

Spatio-temporal multivariate time series (STMTS) are defined by their sequential order and spatio-temporal dependencies, containing valuable information into the dynamics of various systems and processes in communications and networking. Spatio-temporal predictive learning seeks to generate future frames of STMTS by analyzing the available historical frames. The importance of spatio-temporal predictive learning lies mainly in its ability to analyze and model both the spatial correlations and temporal state transitions of the system dynamics. Analyzing and modeling STMTS is a crucial aspect of data mining, providing essential insights and informing decisions across various applications such as multiple-input multiple-output (MIMO) channel prediction [83, 84, 32], mobile traffic analysis [85, 86], network slicing [87, 88], and smart cities [89]. For instance, accurate and timely mobile traffic prediction is needed for the intelligent resource management in network slicing [90], which could mitigate network congestion and improve the quality of services (QoS). Meanwhile, channel prediction can help solve the channel aging issue [83], reduce the pilot overhead [32] and thereby enhance the system performance.

In spatio-temporal predictive learning, understanding both the temporal and spatial dependencies is crucial for enhancing prediction accuracy. Conventional statistical techniques like Historical Average (HA) and Auto-Regressive Integrated Moving Average (ARIMA) often perform poorly in this task, as they were designed to capture only temporal correlations [91]. Additionally, these traditional algorithms inefficiently deal with the non-linear dynamics of time series data. Fortunately, the remarkable advancements in deep learning over the past decade have led researchers to explore numerous data-driven approaches to tackle this challenge, enabling adaptive learning and the modeling of complex non-linear dependencies in spatio-temporal predictive learning.

4.1.1 Literature Review

Initially, pioneering studies in communication and networking were conducted by exploring Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to predict STMTS [92, 93, 84]. In practice, researchers frequently integrate RNNs and CNNs as a hybrid network to simultaneously learn patterns in data over time and space domain [94]. This is because RNNs are good at capturing temporal dependencies, while CNNs focus on identifying spatial dependencies. It has been shown that such hybrid architectures can efficiently capture the spatio-temporal dynamics in time series forecasting. For example, the authors in [95] demonstrated that a neural network with convolutional long-short term memory (ConvLSTM) layers significantly outperform conventional linear regression methods in the spatio-temporal MIMO channel prediction problem. In [96], the authors leveraged CNN and ConvLSTM to predict channel state information (CSI) in high-speed railway networks and verified that the hybrid architecture outperformed the classical CNN and RNN networks in terms of prediction accuracy.

Besides CNNs, Graph Convolutional Networks (GCNs) have recently attracted significant research interest in spatio-temporal forecasting. Unlike CNNs, GCNs are specifically designed to better capture the spatial correlations of graphs in non-Euclidean space. As a result, GCNs are considered as better substitutions for the CNN modules in existing networks to efficiently understand the spatial structure of data. For example, the authors of [97] proposed to model mobile terminals as a time-evolving graph and used GCN to predict the future mobile traffic data. Their experiments demonstrated that the proposed GCN-based model outperformed CNN-based models across various prediction metrics. Likewise, the authors of [98] demonstrated that a GCN backbone achieves superior traffic flow prediction performance compared to CNN backbones.

RNN models, such as LSTM, are widely recognized for their limitations in handling long-range temporal dependencies [99]. To address this issue, Transformers

based on attention mechanisms have recently been proposed [88]. One of the key advantages of Transformers is their ability to capture long-range dependencies and interactions [100]. This capability, stemming from the core attention mechanism within Transformers, is particularly beneficial for time series modeling, leading to significant advancements across various applications. For example, the authors in [101] proposed spatio-temporal Transformer networks (STTN) for traffic flow prediction, demonstrating significant improvements in prediction accuracy compared to ConvLSTM [102]. The authors in [87] employed the attention mechanism in spatio-temporal cellular mobile traffic prediction, achieving a higher prediction accuracy compared with RNN backbones. The encoder-decoder Transformer has shown its advantages in MIMO channel prediction [103], which can accurately predict future channels in parallel. In [104], the authors employed a hybrid network including GCNs and attention mechanism to predict cellular traffic. It has been shown therein that the hybrid architecture effectively leverage the temporal dependencies of cellular traffic and spatial dependencies of the physical network topology.

4.1.2 Motivations and Contributions

The continuous development of spatio-temporal predictive learning for time series has highlighted the essential role of Transformers for the temporal module of hybrid network architectures [105, 106]. The attention model, which forms the foundation of Transformer models, has recently been extensively investigated in time series forecasting [99]. Different variants such as Pyraformer [107], Informer [108], and Reformer [109] primarily focus on reducing computational complexity without compromising prediction performance. Consequently, the vanilla attention model remains widely used in many applications [88, 102, 87, 103, 104, 110, 111], provided the sequence length is manageable. Despite its common use for handling spatially and temporally correlated data, the vanilla attention model was originally designed as a learnable regression kernel based

solely on temporal similarities [105], which potentially limits its capability for spatio-temporal predictive learning. For this reason, the existing approaches often integrate Tranformers models and convolutional networks to jointly capture the spatio-temporal correlations among data frames[88, 103, 112]. Based on this consideration, we argue that addressing this issue can further enhance its efficiency in predicting spatially and temporally correlated data.

In this work, we propose a novel *Crossover Attention (XOA)* model for spatio-temporal predictive learning. It functions as a learnable regression kernel that predicts values by simultaneously considering both spatial and temporal similarities. This feature is particularly appealing for spatio-temporal predictive learning, as it prioritizes input sequences with both spatial and temporal similarities, extracting relevant information for generating future outputs. Our simulation results using synthesized and real-world datasets [89, 113] show that the proposed crossover attention achieves around 1 dB gains on reducing the prediction errors when comparing to the vanilla attention and outperforms recent reference schemes. The novelty of the proposed crossover attention lies on the simple yet effective modification of the vanilla attention mechanism. Extensive ablation simulations verified that performance improvements are achieved by improving the attention mechanism itself rather than employing more traditional attention layers. This demonstrates the effectiveness of the proposed crossover attention mechanism. To conclude, our contributions are twofold:

- *Enhanced Spatio-Temporal Predictive Learning*: We improved the attention model by learning a regression kernel based on both temporal and spatial similarities, which is crucial for applications like channel prediction, traffic prediction, and more. To the best of our knowledge, our work is the first to introduce a spatial attention layer which captures spatial correlations without relying on convolutional networks. The proposed dual attention layer allows the model to prioritize input sequences that are similar in both dimensions, making it more effective for spatio-temporal predictive

learning.

- *Simple but Effective Modification*: The proposed crossover attention is a straightforward modification of the vanilla attention model, making it easy to implement while significantly enhancing performance. This simplicity ensures that it can be readily adopted in various existing frameworks.

4.2 Preliminaries

4.2.1 Spatio-Temporal Predictive Learning

Let $\mathbf{Z}_{1:T} = \{\mathbf{z}_t\}_{t=1}^{T} \in \mathbb{R}^{T \times D_z}$ be a multivariate time series, where each vector $\boldsymbol{z}_t = [z_{t,1}, \ldots, z_{t,i}, \ldots, z_{t,D_z}] \in \mathbb{R}^{D_z}$ represents the variables observed at time t, and $\mathbf{Z}_{t_1:t_2} \in \mathbb{R}^{(t_2-t_1+1) \times D_z}$ represents all values within the time slice $t \in [t_1, t_2]$. This time series exhibits both temporal and spatial correlations. It may be associated with an independent sequence of covariates, denoted by $\mathbf{X}_{1:T} = \{\mathbf{x}_t\}_{t=1}^{T} \in \mathbb{R}^{T \times D_x}$, where each vector $\mathbf{x}_t \in \mathbb{R}^{D_x}$ can contains both dynamic and static domain-specific features.

The goal of spatio-temporal predictive learning is to learn the prediction model

$$\mathbf{Z}_{t+1:t+h} \sim p(\mathbf{Z}_{t+1:t+h} | \mathbf{Z}_{t-w+1:t}, \mathbf{X}_{t-w+1:t+h}; \boldsymbol{\theta}), \tag{4.1}$$

where w represents the maximum length of the moving history window, h represents the prediction horizon, and θ represents the parameters of the model. This model is then employed to predict the future h steps targets $\mathbf{Z}_{t+1:t+h}$ based on a moving window of past w steps observations and the corresponding covariates. In particular, in the absence of covariates, (4.1) simplifies to an auto-regressive prediction model. Additionally, one might want to learn a mapping from input

features to the parameters of the prediction model as

$$\boldsymbol{\theta} = \Psi\left(\mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}; \boldsymbol{\omega}\right), \tag{4.2}$$

where $\Psi\left(\cdot; \boldsymbol{\omega}\right)$ is typically a neural network parameterized by a set of learning parameters $\boldsymbol{\omega}$, such as weights and bias. It is worth noting that $\Psi\left(\cdot; \boldsymbol{\omega}\right)$ is often used to learn the dependency structure among the time series.

4.2.2 Spatio-Temporal Dynamics

In practical applications of spatio-temporal predictive learning, the time series frequently exhibits spatial and temporal correlations. Generally, these spatio-temporal dependencies should be leveraged to enhance the prediction accuracy. To illustrate this, we refer to Fig. 4.1 and Fig. 4.2, which demonstrate the spatio-temporal effects of cellular traffic data in Milan. This data is sourced from a public dataset released by Italia Telecom [113]. The city is partitioned into a grid of 100×100 cells, each measuring 235×235 square meters. A period of 62 days of communication record details (CDRs) was gathered within this area. The original CDRs, aggregated at 10-minute intervals, were resampled at an hourly interval for this demonstration.

Fig. 4.1 presents the hourly aggregated traffic data for a specific cell within the city for the first two weeks of November 2013. It is evident that the traffic data follows a distinct seasonal pattern, demonstrating the temporal correlation of the time series. The daily or weekly traffic for a particular cell is correlated and varies in a similar manner. Additionally, Fig. 4.2 displays a city-wide heatmap of internet activities. It can be observed that the cellular traffic data collected at neighboring cells varies according to their spatial distribution and traffic data collected within the same zone may exhibit similar variations over time, indicating the spatial correlation of the time series.

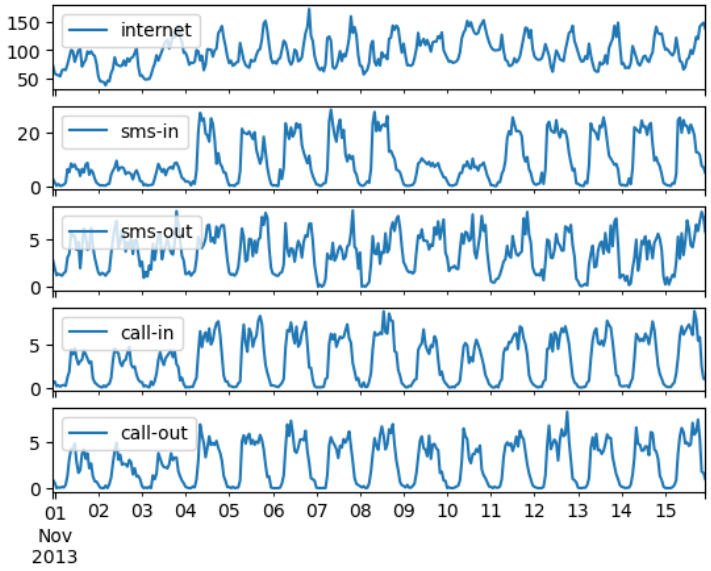

Figure 4.1 Hourly cellular traffic pattern in two weeks of the $(50, 50)$-th cell in Milan, Italy.

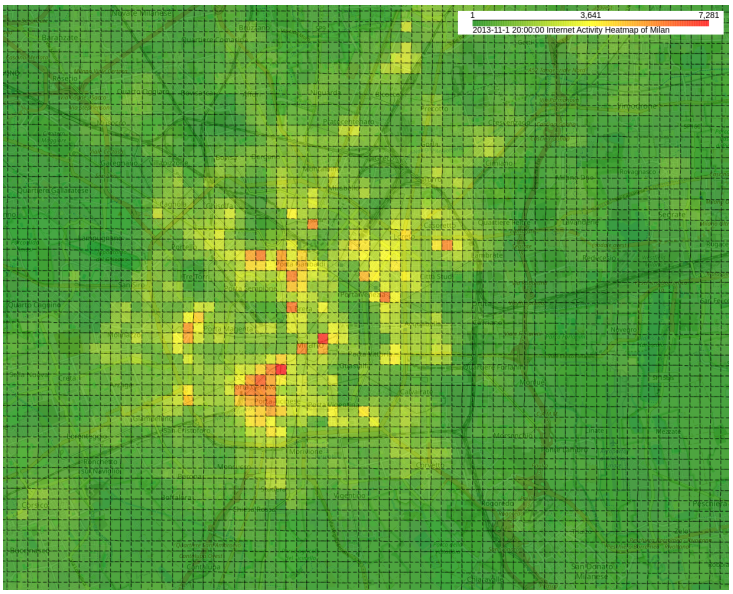

Figure 4.2 Heat map of the internet activities in in Milan, Italy.

4.2.3 Attention Mechanism

The attention mechanism, initially introduced for machine translation in [114], has become a fundamental concept in the deep learning literature. It is designed to capture the long-range dependency structures of the input sequence. The attention mechanism operates as a query-key-value model and typically employs scaled dot-product to calculate the temporal similarities between queries and keys [105]. The outcome is the normalized weighted sum of the training values. The general form of the traditional temporal attention mechanism can be mathematically expressed as

$$\mathtt{A}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\left[\mathtt{softmax}\left(\mathbf{M}_t+\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)\right]\mathbf{V}, \tag{4.3}$$

where $\mathbf{Q}\in\mathbb{R}^{N\times D_k}$, $\mathbf{K}\in\mathbb{R}^{M\times D_k}$, $\mathbf{V}\in\mathbb{R}^{M\times D_v}$ represent queries and training key-value pairs with lengths of N and M, respectively. In addition, $\mathbf{M}_t\in\mathbb{R}^{N\times M}$ denotes the temporal causality mask for masking out similarities that include future frames, which is achieved by adding $-\infty$ to the corresponding components. The symbol $(\cdot)^T$ represents the matrix transpose, and the $\mathtt{softmax}\left(\cdot\right)$ function computes the normalized weights on the last axis of the input tensor. The attention mechanism has been widely used in various spatio-temporal predictive learning tasks, including cellular traffic prediction [83, 84] and MIMO channel prediction [87, 88]. Recent research has adequately confirmed the effectiveness of the attention mechanism in these domains.

4.3 The Proposed Crossover Attention Mechanism

Although the attention mechanism has achieved significant success in spatio-temporal predictive learning, it is not designed for efficiently utilizing cross-

domain correlations. In this section, we present the proposed direct yet effective variant to augment the capabilities of attention models in spatio-temporal predictive learning.

4.3.1 Querying by Temporal Correlations

The vanilla attention mechanism, as outlined in (4.3), can be viewed as a realization of the Naradaya-Watson regression model [105]. Let us denote the temporal view of queries, keys, and values as $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^{N}$, $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{M}$, and $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{M}$, where $\mathbf{q}_i \in \mathbb{R}^{1\times D_k}$, $\mathbf{k}_i \in \mathbb{R}^{1\times D_k}$, and $\mathbf{v}_i \in \mathbb{R}^{1\times D_v}$ are the corresponding spatial vectors. Consequently, the Naradaya-Watson regression model can be expressed as

$$\mathbf{a}_i = \sum\nolimits_{j=1}^{M} \sigma\left(\mathbf{q}_i, \mathbf{k}_j\right) \mathbf{v}_j, \quad \forall i = 1, 2, \ldots, N \tag{4.4}$$

where $\sigma\left(\cdot\right)$ denotes a scalar similarity kernel.

In the setting of spatio-temporal predictive learning as outlined in (4.1), it is appropriate to view key-value pairs as historical covariate-target (features-label) pairs, i.e., $(\mathbf{K}, \mathbf{V}) \triangleq (\mathbf{X}_{1:t}, \mathbf{Z}_{1:t})$. Additionally, queries can be viewed as future h-step covariates, that is, $\mathbf{Q} \triangleq \mathbf{X}_{t+1:t+h}$. The training space contains all observed key-value pairs $\mathcal{D} = \{(\mathbf{k}_i, \mathbf{v}_i)\}_{i=1}^{M}$, and (4.4) forecasts the targets by projecting each $\mathbf{q}$ in $\mathcal{D}$ using the similarity kernel $\sigma\left(\mathbf{q}, \mathbf{k}\right)$.

In the attention mechanism defined in (4.3), the scaled dot-product similarity between vectors is utilized as the similarity kernel, which is

$$\sigma\left(\mathbf{q}_i, \mathbf{k}_j\right) = \texttt{softmax}\left(\frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{D_k}}\right). \tag{4.5}$$

This similarity kernel generates the sample cross-covariance matrix $\mathbf{C} = \mathbf{Q}\mathbf{K}^T \in \mathbb{R}^{N\times M}$ between $\mathbf{Q}$ and $\mathbf{K}$, where the (n, m)-th element $C_{n,m} = \mathrm{Cov}(\mathbf{q}_n, \mathbf{k}_m)$ denotes the covariance between the n-th query and the m-th key in $\mathcal{D}$. Conse-

quently, the resulting attention is computed in the temporal domain. In particular, the sample temporal correlation coefficients are explicitly computed in the self-attention $\mathtt{A}(\mathbf{Z}, \mathbf{Z}, \mathbf{Z})$. Therefore, we refer to $\mathtt{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ as the temporal attention *querying by temporal correlations*.

4.3.2 Querying by Spatial Correlations

Motivated by the preceding analysis, it is suitable to factorize queries and key-value pairs as the spatial view of $\mathbf{Q} = \{\bar{\mathbf{q}}_i\}_{i=1}^{D_k}$, $\mathbf{K} = \{\bar{\mathbf{k}}_i\}_{i=1}^{D_k}$ and $\mathbf{V} = \{\bar{\mathbf{v}}_i\}_{i=1}^{D_v}$, where $\bar{\mathbf{q}}_i \in \mathbb{R}^{N\times 1}$, $\bar{\mathbf{k}}_i \in \mathbb{R}^{M\times 1}$ and $\bar{\mathbf{v}}_i \in \mathbb{R}^{M\times 1}$ are the corresponding temporal vectors. In this case, the regression model can be adjusted as

$$\mathbf{s}_i = \sum_{j=1}^{D_k} \sigma\left(\bar{\mathbf{v}}_i, \bar{\mathbf{k}}_j\right) \bar{\mathbf{q}}_i, \quad \forall i = 1, 2, \ldots, D_v \tag{4.6}$$

with a scalar similarity kernel being the scaled dot-product

$$\sigma\left(\bar{\mathbf{v}}_i, \bar{\mathbf{k}}_j\right) = \mathtt{softmax}\left(\frac{\bar{\mathbf{k}}_j^T \bar{\mathbf{v}}_i}{\sqrt{M}}\right). \tag{4.7}$$

When implementing this regression model as a differentiable neural network layer, it can be expressed as

$$\mathtt{S}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \mathbf{Q}\left[\mathtt{softmax}\left(\mathbf{M}_s + \frac{\mathbf{K}^T\mathbf{V}}{\sqrt{M}}\right)\right], \tag{4.8}$$

Now it is evident that the similarity kernel in (4.8) is computed in the spatial domain, and we employ an optional spatial mask $\mathbf{M}_s$ for masking out unavailable components in the spatial similarity matrix. These unavailable components are typically caused by malfunctioning sensors and can be masked out by adding $-\infty$ before the softmax operation. After the masking, each row of the spatial similarity matrix is normalized using the softmax activation function. This similarity kernel simply generates the sample cross-covariance matrix $\mathbf{G} = \mathbf{K}^T\mathbf{V} \in \mathbb{R}^{D_k\times D_v}$

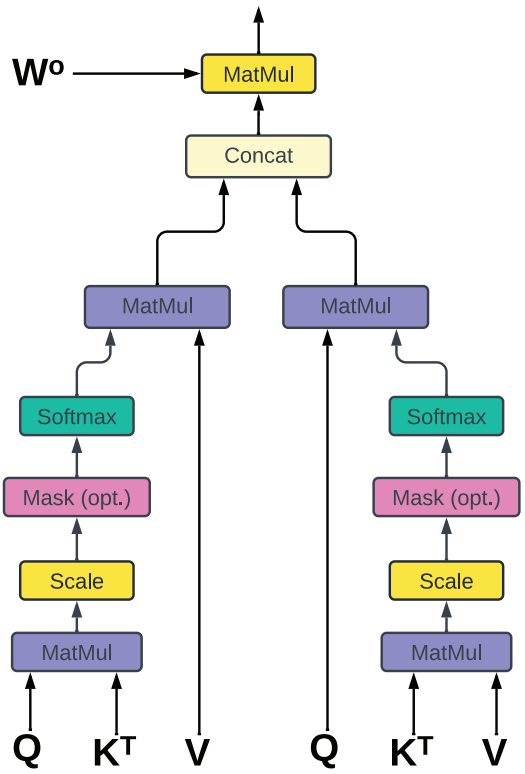

Figure 4.3 Computation graph of the proposed crossover attention mechanism.

between $\mathbf{K}$ and $\mathbf{V}$, in which the (k, v)-th element $C_{k,v} = \mathrm{Cov}(\bar{\mathbf{k}}_k, \bar{\mathbf{v}}_v)$ denotes the covariance between the k-th spatial position of keys and the v-th spatial position of values in $\mathcal{D}$. Specifically, the sample spatial correlation coefficients are explicitly calculated in the self-attention $\mathtt{S}(\mathbf{Z}, \mathbf{Z}, \mathbf{Z})$. Therefore, we refer to $\mathtt{S}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ as the spatial attention *querying by spatial correlations*.

4.3.3 Crossover Attention

In the implementation of a neural network, the results of querying are determined by the similarity kernel $\sigma(\cdot)$, and the kernel is learned implicitly through the projections $\mathbf{Q} = \mathbf{I}\mathbf{W}^q$, $\mathbf{K} = \mathbf{I}\mathbf{W}^k$ and $\mathbf{V} = \mathbf{I}\mathbf{W}^v$, where $\mathbf{I}$ represents the input to the network layer, and $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v$ are learnable projection matrices. To fully utilize the spatio-temporal dependency of the input sequence, the results of

querying from the temporal attention and the proposed spatial attention should be integrated. With this consideration in mind, we design the crossover attention as

$$\texttt{XOA}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\Big[\texttt{A}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right),\texttt{S}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)\Big]\,\mathbf{W}^{O}, \tag{4.9}$$

where $\mathbf{W}^{O}\in\mathbb{R}^{2D_v\times D_v}$ is utilized to integrate the attention values computed by temporal and spatial correlations. In general, (4.9) can be viewed as a component-wise weighted summation of the attention matrices $A(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$ and $S(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$ with the weights controlled by the learnable weight matrix $\boldsymbol{W}^{O}$. The structure of the network implementation of the proposed crossover attention is illustrated in Fig. 4.3. It is important to note that a mask layer is optional before the `softmax` operation if there are some entries that should be masked out for, e.g., computation purpose. The intuition behind the crossover attention design is rather simple: the integration of the cross-domain attentions could help the neural network to learn a more powerful and expressive regression kernel $\sigma(\cdot)$ which jointly considers the temporal and spatial dependencies. As will be shown in the simulation results, the introduced crossover attention outperforms the standard attention mechanism, yielding significantly improved prediction results across diverse applications and datasets.

4.3.4 Complexity Analysis

Analyzing the computational complexity of crossover attention is straightforward. The complexity for the crossover attention described in (4.9) is influenced by both temporal and spatial attentions. For the temporal attention in (4.3), it involves matrix multiplication between an $N\times D_k$ matrix and a $D_k\times M$ matrix, followed by multiplying the resulting $N\times M$ matrix by an $M\times D_v$ matrix. Thus, the complexity of (4.3) is generally $\mathcal{O}\left(NM(D_k+D_v)\right)$. Specifically, for self-attention $\texttt{A}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$, it is $\mathcal{O}\left(T^2D_z\right)$. Similarly, the spatial attention in (4.8) has a

complexity of $\mathcal{O}\left(D_k D_v(N+M)\right)$, and for self-attention $\mathtt{S}\left(\mathbf{Z}_{1:t}, \mathbf{Z}_{1:t}, \mathbf{Z}_{1:t}\right)$, it is $\mathcal{O}\left(TD_z^2\right)$. Therefore, the total complexity of the proposed crossover attention is $\mathcal{O}\left(D_k D_v(N+M) + NM(D_k+D_v) + 2ND_v^2\right)$, and for self-crossover-attention $\mathtt{XOA}\left(\mathbf{Z}_{1:t}, \mathbf{Z}_{1:t}, \mathbf{Z}_{1:t}\right)$, it is $\mathcal{O}\left(T^2 D_z + 2TD_z^2\right)$. For a fixed D_z, it is clear while the computational complexity of the proposed crossover attention is slightly higher than that of traditional attention, it still operates in quadratic time, similar to traditional attention.

Additionally, crossover attention uses the same set of queries, keys, and values for both temporal and spatial attention sub-modules, making it easy to replace the vanilla attention layer with the crossover attention layer. Since the temporal and spatial attentions share the same $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, the number of learnable parameters in the proposed crossover attention consists only of the weights $\boldsymbol{W}^q, \boldsymbol{W}^k, \boldsymbol{W}^v$ and $\boldsymbol{W}^O$. For self-attention, we have $D_k = D_v = D_z$, resulting in $\boldsymbol{W}^q, \boldsymbol{W}^k, \boldsymbol{W}^v \in \mathbb{R}^{D_z \times D_z}$ and $\boldsymbol{W}^O \in \mathbb{R}^{2D_z \times D_z}$. Consequently, the number of parameters in the proposed crossover attention is $5D_z^2$, whereas the number of parameters in traditional attention is $3D_z^2$.

4.4 The Proposed XOATran Architecture

In this section, we introduce a decoder-only Transformer architecture, namely *XOATrans*, which is constructed based on the proposed crossover attention.

4.4.1 Multi-Head Crossover Attention

In practice, transformer models often use multi-head attention mechanism instead of full (single-head) attention [115]. This approach allows the model to jointly attend to information from different representation subspaces at various positions, enhancing the network's expressive power and modeling capabilities. Therefore, we adopt the multi-head pattern for our proposed crossover attention in this

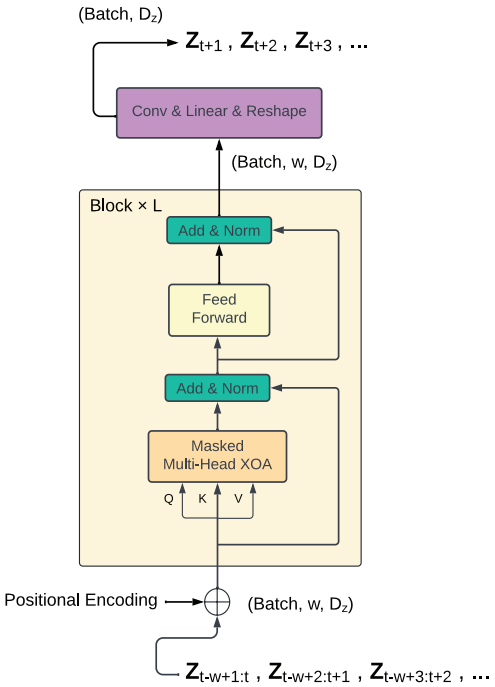

Figure 4.4 Network architecture of XOATran.

paper, and the mathematical model can be expressed as

$$\texttt{MHXOA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_H\right] \mathbf{W}^M, \tag{4.10}$$

where H denotes the total number of heads and each head $\mathbf{h}_i$ is given

$$\mathbf{h}_i = \texttt{XOA}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \tag{4.11}$$

where $\mathbf{W}_i^Q \in \mathcal{R}^{D_k \times d_k}$, $\mathbf{W}_i^K \in \mathcal{R}^{D_k \times d_k}$, $\mathbf{W}_i^V \in \mathcal{R}^{D_v \times d_v}$ and $\mathbf{W}^M \in \mathcal{R}^{Hd_v \times D_v}$ are parameter matrices for the projections. We denote $d_k = D_k/H$, $d_v = D_v/H$ as the dimensionality of each subspace. Because each head operates on a reduced dimension, the overall computational cost remains similar to that of single-head crossover attention with full dimensionality.

4.4.2 Model Architecture

The Transformer model was initially proposed with an encoder-decoder architecture [115]. However, recent designs of Transformers have favored a decoder-only paradigm, especially for large language models (LLMs) [116, 117]. Following this best practice and simplifying the architecture, we adopt the decoder-only paradigm for our proposed XOATran.

As shown in Fig. 4.4, the XOATran comprises a positional encoding layer, L decoder blocks, and an output layer for generating the output with expected shape. Unlike RNNs and CNNs, attention layers lack recurrent states and convolutions, potentially losing relative or absolute positional information during forward passes. To maintain the sequential order, we use a fixed positional encoding similar to [115]. The encoded positional information is added to the input, which is then processed by L decoder blocks. Each decoder block includes a masked multi-head crossover attention layer and a feed-forward network. It should be noted that the output of the decoder blocks will retain the same shape as the input. Following the forward pass, the output of the crossover attention layer will be processed by the `Add&Norm` operation given by,

$$\texttt{Add\&Norm}(\mathbf{I}, \mathbf{P}) = \texttt{Norm}(\mathbf{I} + \mathbf{P}), \tag{4.12}$$

where $\mathbf{I}$ is the input of this layer, $\mathbf{P}$ is the input of previous layer and $\texttt{Norm}(\cdot)$ denotes the layer normalization operation [115]. The `Add&Norm` operation is actually a normalized residual connection which adds the original input to the output of a deeper layer. Residual connections are crucial for the Transformer model architecture. Because they enable effective handling of very deep networks and mitigate the vanishing gradient problem [117]. For the feed-forward network, it can be described as

$$\texttt{FFN}(\mathbf{I}) = \max\left(\mathbf{0}, \mathbf{I}\mathbf{W}_1 + \mathbf{B}_1\right)\mathbf{W}_2 + \mathbf{B}_2. \tag{4.13}$$

The $\texttt{FFN}(\cdot)$ layer is essentially a two-layer perceptron with Rectified Linear Unit (ReLU) activation function and $\mathbf{W}_1$, $\mathbf{W}_1$, $\mathbf{B}_1$ and $\mathbf{B}_1$ being the learnable weights and bias with appropriate shapes. Hence, the output of the decoder block can be described as

$$\mathbf{O}_1 = \texttt{Add\&Norm}(\mathbf{I}, \texttt{MHXOA}(\mathbf{I}, \mathbf{I}, \mathbf{I})), \tag{4.14}$$

$$\mathbf{O}_2 = \texttt{FFN}(\mathbf{O}_1), \tag{4.15}$$

$$\mathbf{O}_3 = \texttt{Add\&Norm}(\mathbf{O}_1, \mathbf{O}_2). \tag{4.16}$$

This process will repeat for L blocks, and the output of the last block will be processed by a CNN layer and/or linear layer to produce the final prediction, depending on the specific application.

4.4.3 Training

In this paper, we adopt a supervising learning approach to train the proposed XOATran. Let the final output of the whole network be denoted as

$$\widehat{\mathbf{Z}}_{t+1:t+h} = \texttt{XOATrans}(\mathbf{Z}_{t-w+1:t}), \tag{4.17}$$

where $\mathbf{Z}_{t-w+1:t}$ is the input sequence with a window length of w, $\widehat{\mathbf{Z}}_{t+1:t+h}$ is the h-step prediction of the ground truth $\mathbf{Z}_{t+1:t+h}$. Then, we compute the loss as

$$\mathcal{L}(\mathcal{D}) = \frac{1}{T} \left\| \texttt{XOATrans}(\mathbf{Z}_{t-w+1:t}) - \mathbf{Z}_{t+1:t+h} \right\|^2, \tag{4.18}$$

$$= \frac{1}{T} \left\| \widehat{\mathbf{Z}}_{t+1:t+h} - \mathbf{Z}_{t+1:t+h} \right\|^2, \tag{4.19}$$

where $\mathcal{D} = \{\mathbf{Z}_{t-w+1:t}, \mathbf{Z}_{t+1:t+h}\}_{t=t_0}^{t_0+T}$ is a dataset of T training pairs. The loss function is the mean squared error (MSE) on the training dataset. We employ the PyTorch framework and the Adam stochastic optimizer [118] to implement and train the model.

4.5 Performance Evaluation

In this section, we present numerical results to verify the effectiveness of the proposed crossover attention in two spatio-temporal predictive learning applications: (i) MIMO channel prediction and (ii) traffic prediction. We also conduct ablation studies by replacing the attention layers inside of recent developed Transformers with our proposed crossover attention, simplifying the performance comparison.

We adopt several metrics for evaluating the prediction accuracy of each model, including mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), normalized root mean square errors (NRMSE), normalized mean squared error (NMSE) and R-squared coefficient of determination (R^2). Note that R^2 provides information about the goodness of a fitting model, and its value normally varies within $[0, 1]$ with 1 indicating the perfect fitting.

In general, all hyper-parameters should be determined based on our available computing resources. While scaling up the model may enhance the network's learning capacity, it can also complicate the training process. Considering these factors, we first establish ranges for all hyper-parameters based on our available computing resources. We then empirically select appropriate values within these ranges, guided by training performance and convergence speed. Notably, for ablation simulations, we follow the same values as specified by the original methods, if applicable.

4.5.1 MIMO Channel Prediction

One important application of spatio-temporal predictive learning in 5G/6G communications systems is the channel prediction. Predicting CSI from the historical observed CSI could help mitigate the channel aging issue and reduce unnecessary pilot overheads. In this case, we perform simulations for MIMO channel prediction by following a similar problem settings as in [32]. Before showing the simulation results, we will briefly introduce the environment setup.

Environment Setup

We perform simulations based on the downlink of a massive multiuser MISO (MU-MISO) system, where a base station (BS) serves N_u single-antenna users. The system operates in time division duplexing (TDD) mode, and the BS has a limited transmit power P_{tot}, N_t transmit antennas and N_f ($0 < N_f \ll N_t$) RF chains. Initially, the users are randomly located within the coverage area of the BS and are assumed to move with constant velocities. Additionally, the channels between the BS and users are time-varying. On the downlink transmission, the CSI acquisition is accomplished via uplink pilot-assisted channel measurement, and multi-user precoding is then adopted to mitigate inter-user interference. The pilot overhead could be prohibitive with limited RF chains in the system. To tackle this issue, one feasible approach is to estimate only partial CSI at each frame and employ channel prediction to recover the full CSI from the historical incomplete observations [19]. Then, the predicted full CSI will be used for selecting antennas for downlink transmission [79, 32]. Due to the Doppler effect and spatially dependent antenna patterns, real propagation environments often exhibit temporal and spatial correlations [11], which can be leveraged to predict future channel states. Because only partial CSIs are available in the history windows, this problem is regarded as a partially observable Markov process (POMDP) which is much challenging than the prediction problems with fully observable channel states [95].

Due to the lack of real data set for this use case, we simulate the propagation environment following the same configuration as in [19, 32], which models the channel evolution by a Gaussian-Markov process with the Jakes' model [65, 67], given by

$$\boldsymbol{h}_{k,t} = \zeta_k \boldsymbol{h}_{k,t-1} + \sqrt{1-\zeta_k^2}\boldsymbol{\Delta}_t, \tag{4.20}$$

where $\zeta_k \in [0, 1]$ represents the temporal correlation coefficient for user k, and

$\mathbf{\Delta}_t \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Sigma})$ is the innovative complex Gaussian i.i.d. in time. The spatially correlated channel vector $\mathbf{\Delta}_t$ follows the Kronecker model [5], and we denote $\alpha \in [0, 1]$ as the spatial correlation coefficient at BS. The value of ζ_k is determined by the maximum Doppler frequency and is inversely proportional to the terminal speed [65], in which $\zeta_k = 1$ represents a static channel and $\zeta_k = 0$ implies that the channel is i.i.d. over time. The fading correlation coefficient can be obtained from Jakes' model given by $\zeta_k = J_0\left(2\pi \frac{v_k f_c}{C} T\right)$, where $J_0(\cdot)$ denotes the zeroth order Bessel function of the first kind, v_k is the speed of user k, C is the speed of light, and T is the frame duration [81]. It should be noted that although an explicit spatio-temporally correlated channel model is adopted here, the algorithm does not have any prior knowledge of the spatio-temporal correlation model. Hence, the prediction algorithm can be applied to any spatio-temporal time series in real-world datasets, as will be verified in Sec. 4.5.2.

In the simulations, the BS is equipped with $N_t = 32$ antennas and $N_f = 16$ RF chains, and $P_{tot} = 100$ watt. The number of users $N_u = 4$, and we adopt a uniform range of speeds from 3.6 km/h to 72 km/h for all users, and the spatial correlation coefficient $\alpha = 0.3$. To train and test the model, the data set is randomly generated with the POMDP introduced in [19, 32]. The process starts by initially selecting an optimal subset of antennas for estimating partial CSI. Subsequently, the full CSI is reconstructed using historical partial estimations. Furthermore, the reconstructed full CSI is used to select antennas for the next frame and the process is repeated. With the POMDP involving, we store at most $1,000,000$ time steps of partial CSIs in a ring buffer, and the stored data is partitioned into a training set and a test set with a ratio of $9:1$. We train the model by using the generated partial CSIs from the training dataset and use the ground-truth CSIs from the same dataset to compute the training losses and update the model's parameters. After updating the parameters at each epoch, we evaluate the model's prediction accuracy using the testing dataset. The number of decoder blocks is $L = 4$, the length of the history window is $w = 24$, and we set the prediction horizon as $h = 1$ since the

interactive process only needs to predict one-step targets.

To verify the effectiveness of proposed crossover attention mechanism, we perform simulations over the following models,

- *XOATran*: The proposed crossover attention enabled Transformer introduced in this paper.

- *Transformer*: The widely adopted decoder-only Transformer model with conventional attention mechanism. In particular, this model is constructed by substituting the crossover attention layer of XOATran with the vanilla attention layer.

- *JCPAS*: The joint channel prediction and antenna selection framework introduced in [32], where the probabilistic prediction networks is based on convolutional network with 24 layers and residual connections. It should be noted that JCPAS is designed to output probabilistic results, and we adjust it to output deterministic results for the ease of comparison.

Numerical Results

Note that for every epoch we test these models on the same test set, and Figs. 4.5- 4.6 illustrates the rolling testing results during the training. Specifically, fig. 4.5 illustrates the NMSE of prediction results during the training process of the aforementioned models. From Fig. 4.5, we can observe that all three models are effective in predicting the full CSI from the history of incomplete observations. Additionally, we can see that Transformer model performs better than the JCPAS using CNNs, as the CNNs struggle to capture the long-range temporal dependencies among the input sequence. Moreover, we can conclude from the figure that the proposed *XOATrans* outperforms the two reference models. Specifically, *XOATrans* achieves a gain of about 1 dB and 2.5 dB compared to *Transformer* and *JCPAS*, respectively. This confirms the effectiveness of the

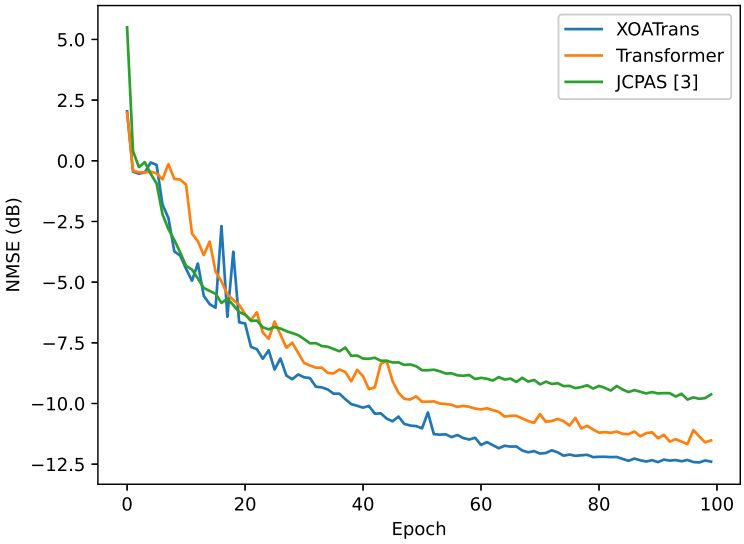

Figure 4.5 NMSE testing results versus training epoch.

proposed crossover attention in capturing both the temporal and spatial structure of the input sequence.

Fig. 4.6 presents the tested sum-spectral efficiency of the aforementioned models. In this figure, the sum-spectral efficiency is computed using the selected antennas, which are chosen based on the predicted full CSI as in [32]. It should be noted that the sum-spectral efficiency can be maximized by selecting the best subset of antennas, and we adopted the norm-based antenna selection algorithm for this purpose. Therefore, accurate channel prediction will eventually result in higher sum-spectral efficiency. From this perspective, we can conclude from this figure that the proposed *XOATrans* outperforms *Transformer* and *JCPAS*. This is because *XOATrans* not only achieves the lowest NMSE but also preserves the ordering information of the norms of channel vectors, which ultimately helps the algorithm to select a better subset of antennas compared to the other two models. While NMSE directly measures the prediction accuracy, the sum-spectral efficiency of antenna selection is presented as an indirect proxy measurement

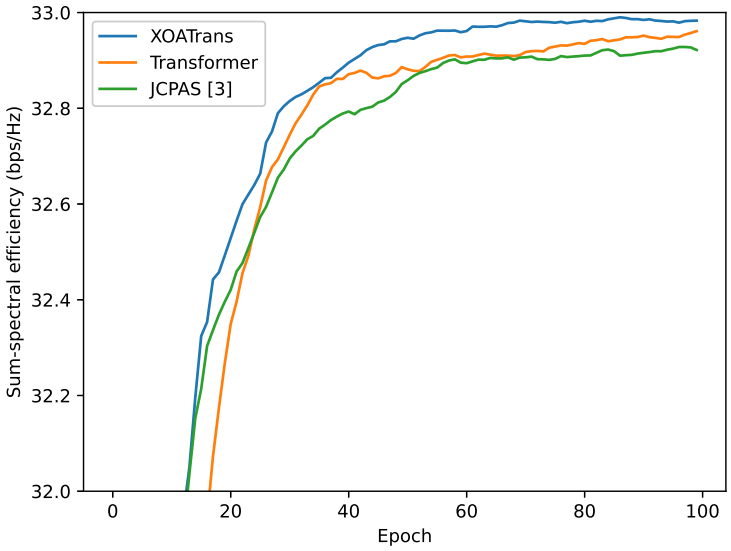

Figure 4.6 Sum-spectral efficiency testing results versus training epoch.

of the prediction accuracy in Fig. 4.6. Therefore, the degree of sum-spectral efficiency gain may appear to be less significant than the NMSE gain. However, we can find from the figure that our proposed *XOATrans* converges much faster than *Transformer*, which uses roughly 30% fewer epochs. These findings further verified the effectiveness of the proposed crossover attention.

4.5.2 Traffic Prediction

In this section, we verify the effectiveness of the proposed crossover attention using two real-world datasets for traffic prediction. Traffic prediction is an important application of spatio-temporal predictive learning. Accurate and timely prediction of the traffics allows more efficient resources allocation and management. Before introducing the results, we will first briefly describe the datasets used in the simulations.

- *Milan* dataset [113]: As shown in Fig. 4.1, this dataset comprises 62 days of cellular mobile traffic data for the city of Milan, Italy. In this dataset, the

entire city area is divided into 100×100 square cells, with cellular traffic data resampled at an hourly granularity. In the experiments, we focus on predicting the number of call-ins for the selected 20×20 cells within this dataset under the same settings in [87].

- *SanDiego* dataset [89]: Compared to the *Milan* dataset, this dataset is significantly larger and is a subset of the LargeST benchmark dataset. It includes road traffic data for over 17,000 road segments and 700 sensors in the area around San Diego, USA. The data is recorded at 5-minute intervals over five years, from 2017 to 2021. In our experiments, we focus on predicting the traffic volumes for all sensors within this dataset.

Competing Algorithms

We conduct ablation experiments by replacing the attention modules in two existing Transformers: (i) ST-Tran-TTB and (ii) STTN. The details of these Transformers can be found in [87], [101], and [89]. For the reader's convenience, we list below the abbreviations of the competing algorithms or neural networks:

- *HA*: The Historical Average (HA) algorithm, which takes the average of its history as the prediction result.
- *HL*: The Historical Last (HL) algorithm, which simply uses the last observation as the future prediction.
- *ARIMA*: The well-known Autoregressive Integrated Moving Average (ARIMA) algorithm, implemented using the *statsmodels* Python library.
- *LSTM*: The long-short term memory (LSTM) neural network for time-series forecasting [93].
- *ConvLSTM*: The convolutional LSTM proposed for STMTS forecasting in [102].

- *STDenseNet*: STDenseNet [92], a prediction model that learns spatio-temporal dependency structures using densely connected CNNs.
- *DCRNN*: The diffusion convolutional recurrent neural network (DCRNN) proposed in [119].
- *AGCRN*: The adaptive graph convolutional recurrent network (AGCRN) proposed in [120].
- *STGCN*: The spatio-temporal graph convolutional networks proposed in [121].
- *ST-Tran-TTB*: ST-Tran, an encoder-decoder Transformer designed for STMTS forecasting with a temporal transformer block (TTB) [87].
- *STTN*: The spatio-temporal Transformer networks (STTN) for spatio-temporal traffic forecasting [101], which integrate GCNs alongside the attention mechanism.
- *ST-Tran-XOA*: Our modified version of ST-Tran, where the attention layers are replaced by our proposed crossover attention layers.
- *STTN-XOA*: Our modified version of STTN, where the attention layers are replaced by our proposed crossover attention layers.

Note that ST-Tran-XOA and STTN-XOA are tested using the same random seeds, hyper-parameters, instructions, and datasets as described in the original papers. This approach allows us to present a clear and straightforward performance comparison to demonstrate the effectiveness of the proposed crossover attention mechanism.

Numerical Result and Discussion

Table 4.1 summarizes the prediction performance comparisons of the competing models for the *Milan* dataset. From this table, we can observe that our proposed

Model	MAE	NRMSE	R^2
HA	18.7226	0.9687	0.4419
ARIMA	17.1895	0.8813	0.6564
LSTM [93]	13.9438	0.6079	0.7802
STDenseNet [92]	12.3168	0.6442	0.7842
ConvLSTM [102]	11.2308	0.5652	0.8097
ST-Tran-TTB [87]	10.3820	0.5521	0.8187
ST-Tran-XOA	**9.9943**	**0.5508**	**0.8196**

Table 4.1 Performance Comparisons for the Milan Dataset.

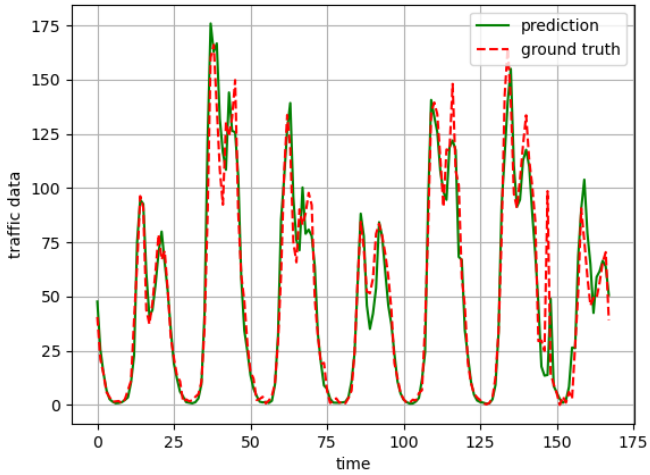

Figure 4.7 The fitness curve of ST-Tran-XOA for the cellular traffic flows in Milan.

crossover attention mechanism achieves the best prediction accuracy in terms of MAE, NRMSE, and R^2. Notably, ST-Tran-XOA attains the highest R^2 score among the seven competing models, indicating that it learns the most fitting model for the spatio-temporal cellular traffic data in the *Milan* dataset. By comparing ST-Tran-XOA with ST-Tran-TTB, we can conclude that the proposed crossover attention mechanism helps the model exploit the spatio-temporal dependencies of the data, resulting in lower prediction errors. In addition to the numerical results presented in Table 4.1, we also depict the predicted results in Fig. 4.7 to illustrate the model fitness of ST-Tran-XOA. As shown in the figure, ST-Tran-XOA can

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE
HL	33.61	50.97	20.77%	57.8	84.92	37.73%	101.74	140.14	76.84%	60.79	87.4	41.88%
LSTM[93]	19.03	30.53	11.81%	25.84	40.87	16.44%	37.63	59.07	25.45%	26.44	41.73	17.20%
DCRNN[119]	17.14	27.47	11.12%	20.99	33.29	13.95%	26.99	42.86	18.67%	21.03	33.37	14.13%
AGCRN[120]	15.71	27.85	11.48%	18.06	31.51	13.06%	21.86	39.44	16.52%	18.09	32.01	13.28%
STGCN[121]	17.45	29.99	12.42%	19.55	33.69	13.68%	23.21	41.23	16.32%	19.67	34.14	13.86%
STTN[101]	16.22	26.22	10.63%	18.76	30.98	12.80%	22.62	39.09	16.14%	18.69	31.11	12.82%
STTN-XOA	**15.57**	**25.48**	**10.01%**	**17.87**	**29.78**	**11.74%**	**21.76**	**37.19**	**14.81%**	**17.85**	**29.74**	**11.72%**

Table 4.2 Performance Comparisons for the SanDiego Dataset.

accurately and smoothly predict the trends and values of future mobile traffic data for a specific cell within the city area, clearly verifying the effectiveness of the proposed crossover attention mechanism.

In addition to the *Milan* dataset, Table 4.2 presents the prediction results of the aforementioned models for the *SanDiego* dataset. From this table, we can see that our proposed crossover attention mechanism outperforms the competing models in terms of prediction errors. Specifically, STTN-XOA achieves the lowest prediction errors among the seven competing models for prediction horizons from 3 to 12. This demonstrates that the proposed crossover attention mechanism is also effective in multi-step prediction tasks. By comparing STTN-XOA with STTN, we can conclude that the model's capability of capturing spatio-temporal dependencies is significantly enhanced by the proposed crossover attention. These results further validate the effectiveness of the proposed crossover attention.

4.6 Conclusion

In this chapter, we investigated the spatio-temporal predictive learning problem, focusing on predicting spatially and temporally correlated time series such as channel states and traffic flows. To efficiently exploit the spatio-temporal correlations, we designed a simple yet effective crossover attention mechanism to help the network understand the spatio-temporal patterns of input data. Experimental results on two popular applications of channel and network traffic predictions using both synthetic and realistic datasets clearly verified the effectiveness of our

proposed crossover attention. Since the proposed crossover attention can be seamlessly integrated into existing models, we believe it offers an attractive method for enhancing the prediction performance of the existing predictive models.

Part II

Model-Free Risk-Aware Learning with Principled Primal Dual Learning

5 Risk-Aware Multi-Agent Packet Routing for LEO Constellations

Prologue. Part I of this dissertation demonstrated the power of a model-based approach for risk-aware decision-making in large-scale communication systems like massive MIMO, where the underlying dynamics can be effectively learned and simulated. However, many emerging large-scale systems are characterized by a degree of complexity, decentralization, and asynchronicity that makes learning an accurate world model intractable. This is particularly true for routing in LEO mega-constellations, where thousands of agents interact in an event-driven manner without a central coordinator or a global clock.

This chapter in part II, therefore, introduces the dissertation's second core paradigm: *model-free risk-aware multi-agent learning*. We shift our focus to the packet routing problem in ultra-dense LEO satellite networks, and propose the `PRIMAL` framework, which is built on the more realistic assumption of asynchronous independent autonomous agents. To manage the risks inherent in decentralized control with local information and uncoordinated action execution, PRIMAL employs a principled primal-dual method that moves beyond optimizing average performance. It learns the full distribution of routing costs to explicitly constrain the tail-end risk via the *Conditional-Value-at-Risk (CVaR)*. This chapter, based on the work in [37], thus presents the model-free fulfillment of the thesis's central goal, showcasing how robust and verifiable risk-awareness can be achieved even in the absence of an explicit *"world model"*.

5.1 Introduction

The rapid development of Low Earth Orbit (LEO) satellite mega-constellations is driving a new era of global connectivity [7, 3, 28]. These networks construct a dynamic space-based backbone of thousands of satellites interconnected by Inter-Satellite Links (ISLs) at altitudes of 500 to 2000 kilometers. Commercial constellations from providers like Starlink, OneWeb, Kuiper, Qianfan and Telesat are now being actively deployed with the ultimate goal of providing ubiquitous high-bandwidth and low-latency internet access to every corner of the globe[122]. In order to realize this vision, a core challenge is designing a robust and adaptive packet routing mechanism for this dynamic infrastructure [28].

However, designing such an algorithm is challenging due to the massive scale, dynamic topology, and significant propagation delays inherent in LEO networks [28, 18, 2]. These characteristics make centralized control with a timely global network view impractical [7, 123]. Consequently, an effective routing algorithm must be decentralized, allowing each satellite to operate *asynchronously* and *independently* using only local information. Furthermore, imbalanced global traffic distribution causes unpredictable congestion, demanding a mechanism for *asynchronous risk-aware packet routing* [3, 28, 18]. Such a mechanism should be able to manage conflicting Quality of Service (QoS) objectives, like latency minimization and load balancing, in a decentralized manner. Addressing this need is the main focus of our work.

Related Works & Limitations. Prior work has explored both traditional and learning-based strategies [22, 3]. Traditional rule-based methods often use static topology snapshots or geographic principles to pre-calculate routes [124, 125, 126, 127]. While some approaches avoid full global knowledge [128], they are fundamentally *risk-oblivious* and fail to handle dynamic events like traffic congestion [129]. To address these limitations, data-driven Deep Reinforcement Learning (DRL) has emerged as a promising direction [35, 130, 36, 131, 132, 29].

While early single-agent RL frameworks exist [35], they suffer from scalability bottlenecks, shifting the focus to decentralized Multi-Agent Reinforcement Learning (MARL) where each satellite acts based on its local observation [130, 133, 36, 132].

However, many existing MARL methods are often misaligned with the physical reality of LEO satellite networks. Approaches based on cooperative MARL paradigms like MAPPO [132, 36] typically enforce action-synchronization by discretizing time into synchronous time-slots, where agents are constrained to make at most one decision per time slot. While individual satellites can achieve high-precision physical clock synchronization, this does not resolve the impracticality of this time-slotted decision paradigm. The core issue is that forcing the naturally event-driven packet routing decision process (i.e., asynchronous packet arrivals or departures) into a rigid time-stepped joint-action model introduces artificial delays and severe scalability bottlenecks, as all agents must wait for a "global tick" before acting [123, 47]. Motivated by this point, asynchronous MARL approaches such as continual DRL with Federated Learning (FedL) and asynchrnous QMIX were proposed to address the issue [29, 50]. However, the absence of synchronized cooperation of all agents can lead to conflicting decisions among agents. This highlights the need for a framework that can manage the risks arising from uncoordinated decentralized actions made by independent agents.

Existing attempts at risk-awareness often rely on heuristic reward shaping. This approach incorporates risk-awareness by engineering complex weighted reward functions that try to balance objectives such as energy budgeting, latency minimization and load balancing [35, 134, 132, 135]. However, such methods lack formal guarantees, and more critically, they require extensive human-effort on trial-and-error based adjusting [21]. A more principled approach is Constrained Reinforcement Learning (CRL). However, recent CRL-based approaches for satellite packet routing like [36], while using a primal-dual method, were risk-myopic to constraining only the average values (neglecting tail-end risks) and relied on

centralized coordinators [136], reintroducing the aforementioned scalability and synchronization problems.

Motivations & Contributions. Our work is motivated by the two critical gaps in existing research. First, the dominant synchronous paradigm in many MARL routing algorithms is poorly suited to LEO networks. They rely on cooperative MARL frameworks that require a centralized coordinator and action-synchronization across all LEOs, which contradicts the inherently asynchronous and event-driven nature of packet routing [48, 47, 137, 49]. Independent MARL approaches, such as [29], remove the need for coordination, but they risk performance degradation as agents may repeatedly interfere with each other's policies. This makes it more difficult to manage diverse and conflicting QoS objectives, reinforcing the need for an asynchronous risk-aware MARL framework.

Second, existing strategies lack robust risk awareness. Many are *risk-oblivious* [35, 132, 135], relying on heuristic reward shaping that lacks formal guarantees and significant human-efforts on coefficient adjusting [21]. Others are *risk-myopic* [36], optimizing only for average performance while ignoring high-impact tail-end events like sudden latency spikes [24].

To address these limitations, we propose `PRIMAL`, a novel *asynchronous* and *risk-aware* MARL framework. `PRIMAL` uses an event-driven design that allows each satellite to act asynchronously based on their local information. Our method includes a principled primal-dual approach which learns the full distribution of the interested risk metrics, and directly constrains worst-case performance risks via distributional reinforcement learning. To summarize this work, our core contributions are:

- We formulate the packet routing problem in satellite networks as an asynchronous MARL problem based on an **event-driven semi-Markov decision process**. Our model operates in continuous time, where each satellite agent acts independently and asynchronously based on its local event-driven timeline and observations. This approach eliminates the unrealistic synchronization

and discrete time-step assumptions of prior RL-based routing methods, leading to a more realistic and efficient routing paradigm.

- We propose a **principled risk-aware routing algorithm using a distributional primal-dual framework**. Instead of learning only the expected costs, our agents learn the full conditional distribution of routing outcomes via quantile regression. By directly constraining the Conditional Value-at-Risk (CVaR) at a given risk level, our algorithm can effectively mitigate tail-end risks, ensuring the routing policy is robust against worst-case performance degradation.

- We develop a **decentralized and synchronization-free scalable learning architecture** without reliance on a centralized coordinator that needs action-synchronization and global state information. This enables extensions such as online federated learning [29] that facilitates scalable and practical online adaption in real-world mega-constellations.

5.2 System Model & Problem Formulation

5.2.1 Network Model

As illustrated in Fig. 5.1, we model the LEO satellite network as a time-varying directed graph $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}(t))$, where the node set $\mathcal{N}$ consists of satellites $\mathcal{N}_s$ and ground stations $\mathcal{N}_g$.

Nodes

The $|\mathcal{N}_s|$ satellites are arranged in a Walker constellation, forming a grid-like topology with Inter-Satellite Links (ISLs) [28]. Each satellite acts as a store-and-forward router with finite buffers. The $|\mathcal{N}_g|$ ground stations act as traffic entry/exit

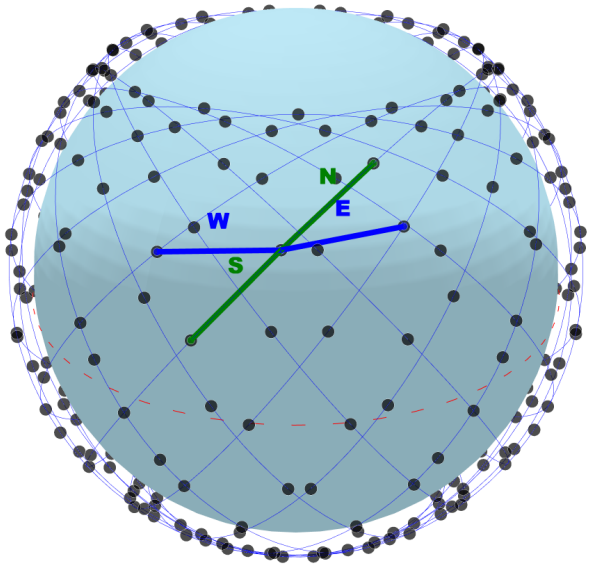

Figure 5.1 Illustration of a Walker-Delta LEO constellation with grid topology, where each satellite connects two intra-plane neighbors (N+S) and two inter-plane neighbors (W+E). Note that we define the four directions w.r.t. the rotating direction of the orbit.

points, connecting to the network via Ground-to-Satellite Links (GSLs) with their respective *access satellites*.

Links, Packets, and Queues

The edge set $\mathcal{E}(t)$ includes dynamic ISLs with Free Space Optical (FSO) Lasers, and GSLs with Ka-Band radio frequency (RF) links, whose availability depends on line-of-sight (LoS) and minimum elevation angles [10]. Data is transmitted as packets, each defined by a tuple $p \triangleq \{s_p, d_p, L_p, \tau_p, \tau_p^{ttl}\} \in \mathcal{P}$, representing its source, destination, size, creation time, and a Time-to-Live (TTL) field initialized to the maximum TTL H, and $\mathcal{P}$ denotes the set of packets. Packets are processed in a First-In-First-Out (FIFO) manner and are dropped if TTL expires or if buffers are full.

5.2.2 Communication and Delay Model

The end-to-end (E2E) delay for a packet traversing the network is the sum of propagation, transmission, and queuing delays at each hop. To model these delays, we begin by defining a single, hop-indexed binary decision variable $x^h_{p,ij} = 1$ if packet $p \in \mathcal{P}$ traverses the link (i, j) as its h-th hop in its path and 0 otherwise, where $h \in \{0, 1, ..., H-1\}$. The delay at hop h depends on the packet's arrival time at the starting node of the hop, and the arrival time is determined by the sum of delays from all previous hops (0 to $h-1$). For a packet p traversing a link (i, j) at time t, the single-hop delay is

$$D^h_{p,ij} = D^P_{ij}(\tau^h_p) + D^T_{ij}(L_p, \tau^h_p) + D^Q_{ij}(\tau^h_p), \tag{5.1}$$

where $D^P_{ij}(t)$ is the propagation delay, $D^T_{ij}(L_p, t)$ is the transmission delay, and $D^Q_{ij}(t)$ is the queuing delay of the sending node i at arrival time $t = \tau^h_p$. The arrival time τ^h_p at the start of hop h can be recursively defined as

$$\tau^h_p = \tau_p + \sum_{k=0}^{h-1} \sum_{(u,v)\in\mathcal{E}(\tau_{p,k})} x^p_{uv,k} \cdot D^p_{uv,k} \tag{5.2}$$

Now we are ready to introduce the models of each delay component.

Propagation Delay

The propagation delay $D^P_{ij}(t)$ is the time needed for a signal to travel from node i to node j via link (i, j) at time t. It is determined by the Euclidean distance $d_{ij}(t)$ between the nodes at time t, as well as the speed of light κ_c, which is given by

$$D^P_{ij}(t) = \frac{d_{ij}(t)}{\kappa_c}. \tag{5.3}$$

Transmission Delay

The transmission delay $D_{ij}^{T}(L_p, t)$ is the time required to send all the L_p bits of packet p onto communication link (i, j) at time t. It is a function of the packet size L_p and the link's achievable data rate $R_{ij}(t)$. For GSLs operating in the Ka-Band, the data rate is modeled by the Shannon-Hartley theorem, which depends on the link's bandwidth B_{ij} and its Signal-to-Noise Ratio (SNR):

$$R_{ij}^{GSL}(t) = B_{ij} \log_2(1 + \text{SNR}_{ij}(t)). \tag{5.4}$$

The $\text{SNR}_{ij}(t)$ is determined by [132]:

$$\text{SNR}_{ij}(t) = \frac{P_{ij}^{T} G_{ij}^{T} G_{ij}^{R}}{L_{ij}(t) \kappa_B T_{ij} B_{ij}}, \tag{5.5}$$

where P_{ij}^{T} is the antenna transmission power, G_{ij}^{T} and G_{ij}^{R} are the transmitter and receiver antenna gains, κ_B is the Boltzmann's constant, L_{ij} is the Free Space Path Loss (FSPL) and T_{ij} is the system noise temperature in Kelvin. L_{ij} is a function of the distance $d_{ij}(t)$ and carrier frequency of Ka-Band f_c:

$$L_{ij}(t) = \left(\frac{4\pi d_{ij}(t) f_c}{\kappa_c} \right)^2. \tag{5.6}$$

For FSO Laser based ISLs, which rely on FSO communication, the data rate is modeled differently [128]:

$$R_{ij}^{ISL}(t) = \frac{\tilde{B}_{ij}}{2} \log_2 \left(1 + \kappa_1 \cdot e^{-\kappa_2 \cdot d_{ij}(t)} \right), \tag{5.7}$$

where $\tilde{B}_{ij}$ is the optical bandwidth. The parameters κ_1 and κ_2 are related to the average optical SNR and attenuation conditions [138, 139]. Consequently, the

transmission delay for a packet p of size L_p over link (i, j) is given by

$$D_{ij}^{T}(L_p, t) = \frac{L_p}{R_{ij}(t)}, \tag{5.8}$$

where $R_{ij}(t)$ is either $R_{ij}^{GSL}(t)$ or $R_{ij}^{ISL}(t)$ depending on the link type.

Queuing Delay

The queuing delay $D_{ij}^{Q}(t)$ is the time a packet spends waiting in an output buffer before its transmission begins. In a FIFO output queue, this delay is the sum of the transmission delays of all preceding packets in the queue for the same outgoing link. If packet p arrives at node i at time t and is routed to next-hop j, its queuing delay can be approximately calculated as

$$D_{ij}^{Q}(t) = \sum_{q \in \mathcal{P}_{ij}(t)} \frac{L_q}{R_{ij}(t)}, \tag{5.9}$$

where $\mathcal{P}_{ij}(t)$ is the set of packets already in the output queue for link (i, j) at the time t. The queuing delay is a direct indicator of local congestion, as a congested node inherently results in longer packet queue. Consequently, the accumulated queuing delay of a packet's journey can serve as an ideal metric for evaluating the network's load balancing performance.

5.2.3 Problem Formulation

The packet routing problem can be formulated as a large-scale non-linear integer programming problem. The main objective is to find an optimal routing policy, defined by the set of $|\mathcal{P}| \times |\mathcal{E}| \times H$ decision variables $\{x_{p,ij}^{h}\}$, that minimizes the total E2E delay of each packet

$$D_p = \sum_{h=0}^{H-1} \sum_{(i,j) \in \mathcal{E}(\tau_p^h)} x_{p,ij}^{h} \cdot D_{p,ij}^{h}, \tag{5.10}$$

while subjects to several fundamental constraints that define valid routing paths.

First, a valid path must be contiguous. The destination node of any given hop must serve as the source node for the subsequent hop. This path connectivity constraint is enforced for all non-terminal nodes:

$$\sum_{i \in \mathcal{N}_k} x_{ik,p}^h = \sum_{j \in \mathcal{N}_k} x_{kj,p}^{h+1} \quad \forall p \in \mathcal{P}, k \in \mathcal{S}, h < H-1. \tag{C1}$$

In addition, each packet's journey must start at its source ground station s_p and eventually terminate at its destination ground station d_p. The source and destination constraints for packet delivery are

$$\sum_{j \in \mathcal{N}_{s_p}} x_{s_p,j,p}^1 = 1 \quad \text{and} \quad \sum_{h=0}^{H-1} \sum_{i \in \mathcal{N}_{d_p}} x_{i,d_p,p}^h = 1 \quad \forall p \in \mathcal{P}. \tag{C2}$$

Moreover, to ensure that the path is simple and loop-free within a hop index, a packet can traverse at most one link for any given hop h:

$$\sum_{(i,j) \in \mathcal{E}(\tau_p^h)} x_{p,ij}^h \leq 1 \quad \forall p \in \mathcal{P}, h \in \{0, 1, ..., H-1\} \tag{C3}$$

To manage congestion, we add a constraint on the maximum accumulated queuing delay:

$$D_p^Q = \sum_{h,(i,j)} x_{p,ij}^h \cdot D_{ij}^Q(\tau_p^h) \leq D_{max}^Q, \quad \forall p \in \mathcal{P} \tag{C4}$$

where D_{max}^Q is a predefined threshold. The full problem is:

$$\textbf{P1:} \min_{\{x_{p,ij}^h\}} \quad \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} D_p \quad \text{s.t. (C1), (C2), (C3), (C4)} \tag{5.11}$$

This problem is intractable for decentralized solutions due to its non-linear and interdependent nature (e.g., queuing delay couples all routing decisions) and the

massive scale of the network, which makes centralized solvers infeasible [140]. This motivates our decentralized and scalable learning framework.

5.3 Principled Risk-Aware Independent Multi-Agent Learning

To overcome the limitations of synchronized MARL models discussed earlier [36, 132], we adopt an asynchronous event-driven perspective. We model the trajectory of a single packet as a *Partially-Observed Constrained Semi-Markov Decision Process* (POCSMDP), where satellite routers are independent decision-makers. This packet-centric view defines a finite-horizon learning episode for each packet's journey.

5.3.1 Event-Driven Semi-Markov Decision Process

The POCSMDP for a packet p is defined by the tuple,

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \{c_k\}, \mathcal{O}, H, \gamma_r, \gamma_c \rangle_p, \tag{5.12}$$

where:

- $\mathcal{S}$ is the global network state space. A state $s_h \in \mathcal{S}$ is a snapshot of the network's physical status and packet-specific information at the h-th hop.
- $\mathcal{A}$ is the packet routing action space at a satellite, i.e., the set of four outgoing ISLs (NSWE).
- $\mathcal{T}(s', \tau, |s, a)$ is the transition probability to state s' after a variable duration τ (the single-hop delay), making this a *semi-Markov* process.
- $\mathcal{O}$ is the observation function yielding a local observation $o \sim \mathcal{O}(\cdot|s)$, which includes packet state and local node/neighbor statistics, making the

process *partially observable*.

- $r(o, a, o')$ and $\{c_k(o, a, o')\}_{k=1}^{K}$ are the reward (primary objective) and QoS cost functions (e.g., load balancing) for a given state transition.

- $\gamma_r, \gamma_c \in (0, 1]$ are discount factors.

This event-driven formulation ensures that satellites react to packet arrivals in real-time based on current local information. While each packet defines a conceptual learning episode, the physical agents are the satellites. For scalability, we employ parameter sharing, where all satellites use a single, homogeneous policy $\pi_\theta(a|o)$, stored locally but shared across the network, enabling distributed and asynchronous execution.

5.3.2 Maximum Entropy Constrained Reinforcement Learning

Following the event-driven POCSMDP framework, we can solve the routing problem **P1** via CRL [141]. Since each satellite acts independently without coordination, maintaining a certain level of policy stochasticity is beneficial for exploration and for mitigating the non-stationary environment issue arising from the concurrent learning of other agents. This can be achieved by incorporating an additional constraint on the expected policy entropy during learning, known as maximum entropy RL [43]. For each packet p, we are interested in the two variables, the reward-return $Z_\pi^r = \sum_{h=0}^{H-1} \gamma_r^h r(o_h, a_h, o_{h+1})$, and the cost-return $Z_\pi^{c_k} = \sum_{h=0}^{H-1} \gamma_c^h c_k(o_h, a_h, o_{h+1})$, all induced by a fixed policy π. The overall objective is to find the optimal policy that solves the following constrained optimization

problem:

$$\textbf{P2:} \quad \max_{\pi} \quad \mathcal{J}_r(\pi) = \mathbb{E}\left[Z_\pi^r\right] \tag{5.13a}$$

$$\text{s.t.} \quad \mathcal{J}_{c_k}(\pi) \leq D_k, \quad \forall k \in \{1, \ldots, K\} \tag{5.13b}$$

$$\mathcal{H}\left(\pi(\cdot|o_h)\right) \geq \bar{\mathcal{H}}, \quad \forall h \in \{1, \ldots, H\} \tag{5.13c}$$

where $\mathcal{H}\left(\pi(\cdot|o_h)\right) = \mathbb{E}_{a\sim\pi(\cdot|o)}\left[-\log \pi(a|o)\right]$ denotes the entropy of the policy, and $\bar{\mathcal{H}}$ is the desired minimum expected policy entropy. The term $\mathcal{J}_{C_k}(\pi) \leq D_k$ represents a placeholder for the k-th QoS constraint, which can be a constraint on the expected cost-return as

$$\mathcal{J}_{c_k}(\pi) = \mathbb{E}\left[Z_\pi^{c_k}\right] \leq D_k, \tag{5.14}$$

where D_k is the desired cost threshold.

In addition to constraining the expected costs, we further consider risk-averse probabilistic constraints based on Conditional Value-at-Risk (CVaR). Specifically, the Value-at-Risk (VaR) at risk level $\epsilon_k \in (0, 1)$ is the $(1 - \epsilon_k)$-quantile of the cost distribution defined as

$$\mathrm{VaR}_{\epsilon_k}(Z_\pi^{c_k}) = \inf\{z \in \mathbb{R} : F_{Z_\pi^{c_k}}(z) \geq 1 - \epsilon_k\}, \tag{5.15}$$

where $F_{Z_\pi^{c_k}}$ is the Cumulative Distribution Function (CDF) of $Z_\pi^{c_k}$. The CVaR is then defined as the expected cost in the worst ϵ_k tail of the distribution [24]:

$$\mathrm{CVaR}_{\epsilon_k}(Z_\pi^{c_k}) = \mathbb{E}_\pi\Big[Z_\pi^{c_k}\Big|Z_\pi^{c_k} \geq \mathrm{VaR}_{\epsilon_k}(Z_\pi^{c_k})\Big], \tag{5.16}$$

Then, $\mathcal{J}_{c_k}(\pi) \leq D_k$ can be a risk-averse constraint as

$$\mathcal{J}_{c_k}(\pi) = \mathrm{CVaR}_{\epsilon_k}\left(Z_\pi^{c_k}\right) \leq D_k, \tag{5.17}$$

which requires that the expected cost-return $Z_\pi^{c_k}$, conditioned on being in the worst ϵ_k-percentile of outcomes, does not exceed the threshold D_k. This allows for direct control over worst-case scenarios, moving beyond simple average performance.

Note that traditional methods for solving **P2** often rely on reward engineering, where a hand-crafted reward function is designed as a weighted sum of the main objective $\mathcal{J}_r$ and the cost components $\mathcal{J}_{c_k}$. This approach relaxes the constraints by incorporating them as penalties into the objective function, using a set of fixed coefficients. Though effective, it requires manually adjusting these coefficients to balance multiple, often conflicting objectives, which is known to be notoriously challenging in practice [21].

In contrast, primal-dual learning introduces a more principled alternative that directly solves the CRL problem by learning the best multipliers for the constraints [142, 143, 24]. Being reward-agnostic and requiring no prior knowledge, the approach can handle a wide range of general constraints, which eliminates the significant human-efforts required on adjusting the penalty coefficients.

Hence, we propose the **P**rincipled **R**isk-aware **I**ndependent **M**ulti-**A**gent **L**earning (**PRIMAL**) framework to solve **P2** via primal-dual learning. Our approach first transforms the constrained problem into an unconstrained one via Lagrange multipliers. We then solve the resulting Lagrangian dual problem by extending the Soft Actor-Critic (SAC) algorithm to our multi-agent setting with discrete actions [43, 44]. We now introduce two variants of our algorithm: **PRIMAL-Avg**, which handles constraints on expected costs, and **PRIMAL-CVaR**, which addresses constraints on worst-case costs. Both the two variants work asynchronously and independently with only limited local information, ensuring scalability and robustness in large-scale systems.

5.3.3 PRIMAL-Avg: Routing with Expected Cost Constraints

To solve the constrained optimization problem **P2** with expectation constraints, we define the Lagrangian as:

$$\mathcal{L}(\pi, \boldsymbol{\lambda}, \alpha) = \mathcal{J}_r(\pi) + \alpha \left(\mathbb{E}_\pi[\mathcal{H}(\pi)] - \bar{\mathcal{H}} \right) - \sum_{k=1}^{K} \lambda_k \left(\mathcal{J}_{c_k}(\pi) - D_k \right), \tag{5.18}$$

where $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_K\}$ with $\lambda_k \geq 0$ and $\alpha \geq 0$ are the Lagrange multipliers for the K cost constraints and entropy constraint, respectively. This Lagrangian function combines the original objective with the constraints into a single equation. The core idea of the primal-dual method is to transform the original constrained problem, known as the primal problem, into an equivalent dual problem. We first define the dual function $g(\lambda, \alpha) = \max_\pi \min_{\boldsymbol{\lambda} \geq \mathbf{0}, \alpha \geq 0} \mathcal{L}(\pi, \boldsymbol{\lambda}, \alpha)$ as the maximum value of the Lagrangian with respect to the policy π for a fixed set of multipliers. The dual problem then involves finding the multipliers that *minimize* the dual function, i.e., $\min_{\lambda \geq 0, \alpha \geq 0} g(\lambda, \alpha)$. Note that primal-dual CRL has been proved to have strong duality for single-agent fully observable RL [21]. However, the constrained and partially observable MARL problem of our case is fundamentally more challenging and highly nonconvex, such that there exist none known strong duality guarantees [144]. Despite this fact, primal-dual approach still serves as a feasible and principled way to solve the problem approximately.

We approach this using an actor-critic framework with function approximation, i.e., neural networks, for the policy (actor) and the state-action value functions (Q-functions, as the critics). In the maximum entropy framework, the soft reward critic Q^r_ϕ, parameterized by ϕ, is defined as the expected sum of future rewards and entropy bonuses after taking action a in observation o and then following policy π_θ thereafter:

$$Q^r_\phi(o, a) = \mathbb{E}_{\pi_\theta} \left[\sum_{h=0}^{H-1} \gamma_r^h r(o_h, a_h, o_{h+1}) + \alpha \mathcal{H}(\pi(\cdot|o_h)) \Big| o_0 = o, a_0 = a \right], \tag{5.19}$$

Analogously, we have the k-th cost critic $Q^c_{\psi_k}$ as

$$Q^c_{\psi_k}(o,a) = \mathop{\mathbb{E}}_{\pi_\theta}\left[\sum_{h=0}^{H-1}\gamma_c^h c_k(o_h,a_h,o_{h+1})\Big|o_0=o,a_0=a\right], \tag{5.20}$$

with parameters ψ_k. Then, we have the recursive Bellman equations as

$$Q^r_\phi(o,a) = r(o,a,o')\gamma_r \mathbb{E}_{a'\sim\pi_\theta(\cdot|o')}\left[Q^r_\phi(o',a') - \alpha\log\pi(a'|o')\right], \tag{5.21}$$

$$Q^c_{\psi_k}(o,a) = c_k(o,a,o') + \gamma_c \mathbb{E}_{a'\sim\pi_\theta(\cdot|o')}\left[Q^c_{\psi_k}(o',a')\right] \tag{5.22}$$

For discrete actions, we use $Q^r_\phi(o)\in\mathbb{R}^{|\mathcal{A}|\times 1}$, $Q^c_\psi(o)\in\mathbb{R}^{|\mathcal{A}|\times 1}$ and $\pi_\theta(o)\in\mathbb{R}^{|\mathcal{A}|\times 1}$ to denote the per-action outputs. The reward and cost critics can be learned by comparing their predictions to a Temporal Difference (TD) target calculated from the experience $(o,a,r,\{c_k\},o')\sim\mathcal{D}$ drawn from a replay buffer $\mathcal{D}$. Formally, the TD targets can be computed following the Bellman equations as

$$y^r_{\phi'} = r + \gamma_r \pi_\theta^\top(o')\left(Q^r_{\phi'}(o') - \alpha\log\pi(o')\right), \tag{5.23}$$

$$y^c_{\psi'_k} = c_k + \gamma_c \pi_\theta^\top(o') Q^c_{\psi'_k}(o'), \tag{5.24}$$

where $(\cdot)^\top$ denotes the matrix transpose. In practice, these targets are estimated via the corresponding target networks $Q^r_{\phi'}$ and $Q^c_{\psi'_k}$ with mirrored parameters ϕ' and $\{\psi'_k\}_{k=1}^K$. The SAC framework optimizes the reward and cost critics according to their TD errors as

$$\mathcal{L}_\phi = \mathbb{E}_{(o,a,r,o')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q^r_\phi(o,a) - y^r_{\phi'}\right)^2\right], \tag{5.25}$$

$$\mathcal{L}_{\psi_k} = \mathbb{E}_{(o,a,c_k,o')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q^c_{\psi_k}(o,a) - y^c_{\psi'_k}\right)^2\right], \tag{5.26}$$

For the policy update, the policy parameters θ are optmized by minimizing the KL-divergence between the policy and a target distribution derived from the Q-function. Specifically, the policy is updated towards the exponential of

the Lagrangian-based state-action values [44]. The objective for the actor is to minimize the following loss function[43]:

$$\mathcal{L}_\theta = \mathop{\mathbb{E}}_{o\sim\mathcal{D}}\left[\pi_\theta^\top(o)\Big(\alpha\log\pi_\theta(o) - Q_\phi^r(o) + \sum_k \lambda_k Q_{\psi_k}^c(o)\Big)\right] \tag{5.27}$$

This update encourages the policy to select actions that have high reward Q-values and low cost Q-values (weighted by the Lagrange multipliers λ_k), while also maintaining high entropy to facilitate exploration.

The Lagrange multipliers, which are crucial for enforcing the constraints, are updated to minimize the Lagrangian. This results in simple update rules where each multiplier is adjusted based on the extent of its corresponding constraint violation.

$$\mathcal{L}_{\lambda_k} = \mathop{\mathbb{E}}_{o\sim\mathcal{D}}\left[\lambda_k\left(\pi_\theta^\top(o)Q_{\psi_k}^c(o) - D_k\right)\right], \tag{5.28}$$

$$\mathcal{L}_\alpha = \mathop{\mathbb{E}}_{o\sim\mathcal{D}}\left[\alpha\left(-\pi_\theta^\top(o)\log\pi_\theta(o) - \bar{\mathcal{H}}\right)\right]. \tag{5.29}$$

The update for λ_k in (5.28) increases the multiplier if the estimated cost exceeds the threshold D_k, thereby strengthening the penalty on costly actions in the actor's objective. Conversely, it decreases if the constraint is satisfied. Similarly, the entropy multiplier α is adjusted in (5.29) to ensure the policy's entropy remains close to the target level $\bar{\mathcal{H}}$.

5.3.4 PRIMAL-CVaR: Routing with Worst-Case Cost Constraints

While `PRIMAL-Avg` ensures that QoS constraints are met on average, it remains oblivious to low-probability but high-impact events, such as sudden latency spikes that cause cascading network congestion. For mission-critical infrastructure like LEO networks, such tail-end risks are unacceptable. To address this, we introduce `PRIMAL-CVaR`, a variant that directly controls the tail-end risk of the cost

distribution by satisfying the CVaR constraints defined in (5.17). This requires moving beyond learning the mere expectation of the cost-return.

To be aware of the worst-case cost values, the critic should be able to learn the full conditional distribution of the cost-return $Z_\pi^{c_k}$. We achieve this using a powerful distributional RL method known as Implicit Quantile Networks (IQN) [143]. Unlike traditional methods that learn the expected value (i.e., the mean) of a return, IQN aims to capture its full distribution. Its core idea is to approximate the quantile function (the inverse of the CDF) by learning a mapping from a probability $\zeta \in [0, 1]$ to the corresponding return value. This enables the network to implicitly model the entire distribution by being able to estimate any of its quantiles.

Specifically, we adapt IQN for discrete multi-agent SAC. This is realized by a network, $Q^c_{\psi_k}(o, a, \zeta)$, which takes a sample $\zeta \sim U(0, 1)$ as an additional input to generate a value for that specific quantile. This technique provides a far richer and more accurate representation of the cost-return distribution, forming a solid foundation for risk-aware control [142, 141]. The IQN-based cost critic is trained by minimizing the quantile regression loss, guided by the distributional Bellman equation:

$$\begin{aligned} Z_\pi^{c_k}(o, a) &= \sum_{h=0}^{H-1} \gamma_c^h c_k(o_h, a_h, o_{h+1}) | o_0 = o, a_0 = a \\ &\overset{D}{=} c_k(o, a, o') + \gamma_c Z_\pi^{c_k}(o', a'), \end{aligned} \tag{5.30}$$

where $o' \sim P(\cdot|o, a)$, $a' \sim \pi_\theta(\cdot|o')$ and $\overset{D}{=}$ denotes equality in distribution. To leverage this for network training, we employ a specific sampling strategy for each transition (o, a, c_k, o') drawn from the replay buffer $\mathcal{D}$. First, we sample N quantile fractions from the standard uniform distribution, i.e., $\{\zeta_i\}_{i=1}^N \sim \mathcal{U}(0, 1)$, to evaluate the current quantile estimates $Q^c_{\psi_k}(o, a, \zeta_i)$. Second, we sample another N' quantile fractions $\{\zeta'_j\}_{j=1}^{N'} \sim \mathcal{U}(0, 1)$ to construct the TD target using

the target network $Q^c_{\psi'_k}$. For each pair (i, j), the TD error is:

$$\delta_{ij} = \left(c_k + \gamma_c \pi_\theta^\top(o') Q^c_{\psi'_k}(o', \zeta'_j)\right) - Q^c_{\psi_k}(o, a, \zeta_i), \tag{5.31}$$

where $Q^c_{\psi'_k}(o', \zeta'_j) \in \mathbb{R}^{|\mathcal{A}|\times 1}$ outputs the per-action per-sample quantile Q-values for discrete action spaces. The parameters ψ_k of the cost critic are then updated by minimizing the total quantile Huber loss, averaged over all samples:

$$\mathcal{L}_{\psi_k} = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \mathbb{E}\left[\rho_{\zeta_i}(\delta_{ij})\right], \tag{5.32}$$

where $\rho_{\zeta_i}(u) = |\zeta_i - \mathbb{I}(u < 0)| \mathcal{L}_{\text{Huber}}(u)$ is the quantile Huber loss [143]. This asymmetrically weighted loss penalizes over-estimation and under-estimation differently for each quantile ζ_i, which forces the critic to learn an accurate representation of the entire distribution [24].

With a fully characterized cost distribution, we can directly estimate the worst-case costs. To estimate the $\mathrm{CVaR}_{\epsilon_k}$ at a given risk level ϵ_k for a state-action pair, $\Gamma_{\epsilon_k}(o, a) = \mathrm{CVaR}_{\epsilon_k}(Z^{c_k}_\pi(o, a))$, we draw N^k i.i.d. samples $\{\zeta_m\}_{m=1}^{N^k}$ from the re-parametrized uniform distribution $\mathcal{U}(1 - \epsilon_k, 1)$. The CVaR is then approximated by averaging the critic's output for these tail-end quantile fractions:

$$\Gamma_{\epsilon_k}(o, a) \approx \frac{1}{N^k} \sum_{m=1}^{N^k} Q^c_{\psi_k}(o, a, \zeta_m). \tag{5.33}$$

This estimate is then used to guide the actor, where the actor's objective function is modified to incorporate the CVaR estimate, averaged over the policy's action distribution:

$$\mathcal{L}_\theta = \mathop{\mathbb{E}}_{o\sim\mathcal{D}}\left[\pi_\theta^\top(o)\left(\alpha \log \pi_\theta(o) - Q^r_\phi(o) + \sum_k \lambda_k \Gamma_{\epsilon_k}(o)\right)\right], \tag{5.34}$$

where $\Gamma_{\epsilon_k}(o) \in \mathbb{R}^{|\mathcal{A}|\times 1}$ is a vector of per-action $\Gamma_{\epsilon_k}(o, a)$ estimates for discrete

Algorithm 11: The PRIMAL Algorithm

Input: Risk mode $\texttt{M} \in \{\texttt{Avg}, \texttt{CVaR}\}$, $\eta \in (0, 1]$

Initialize π_θ, Q^r_ϕ, $\{Q^c_{\psi_k}\}_{k=1}^K$ and their targets ϕ', $\{\psi'_k\}$;
$\phi' \leftarrow \phi$ and $\psi'_k \leftarrow \psi_k$;
Initialize multipliers $\boldsymbol{\lambda}$, α, and shared replay buffer $\mathcal{D}$;
for *each event* e **do**
  if e *is a packet arrival event* (p, h) **then**
    Get packet p observation $o_h \sim \mathcal{O}(\cdot|s_h)$;
    Execute action $a_h \sim \pi_\theta(\cdot|o_h)$;
  else if e *is an action completion event* $(o', r, \{c_k\})$ **then**
    Collect transition $(o, a, r, \{c_k\}, o')$ and add to $\mathcal{D}$;
  for *each training step* **do**
    Sample a mini-batch $(o, a, r, \{c_k\}, o') \sim \mathcal{D}$;
    Update reward critic ϕ via (5.25);
    if $\texttt{M} = \texttt{Avg}$ **then**
      Update expected cost critic ψ_k via (5.26);
      Update actor θ via (5.27);
      Update cost multipliers $\boldsymbol{\lambda}$ via (5.28);
    else if $\texttt{M} = \texttt{CVaR}$ **then**
      Update distributional cost critic ψ_k via (5.32);
      Update actor θ via (5.34);
      Update cost multipliers $\boldsymbol{\lambda}$ via (5.35);
    Update entropy multiplier α via (5.29);
    Soft update reward target: $\phi' \leftarrow \eta\phi + (1-\eta)\phi'$;
    Soft update cost target: $\psi'_k \leftarrow \eta\psi_k + (1-\eta)\psi'_k$;
  end
end

action space. The update for the Lagrange multiplier λ_k is also adjusted to reflect the CVaR constraint violation, ensuring the policy is pushed towards risk-averse behavior:

$$\mathcal{L}_{\lambda_k} = \lambda_k \mathop{\mathbb{E}}_{o\sim\mathcal{D}}\Big[\pi_\theta^\top(o)\Gamma_{\epsilon_k}(o) - D_k\Big]. \tag{5.35}$$

Note that the updates for the reward critic and the entropy multiplier α remain the same as in **PRIMAL-Avg**. Clearly, by replacing the standard cost critic with a

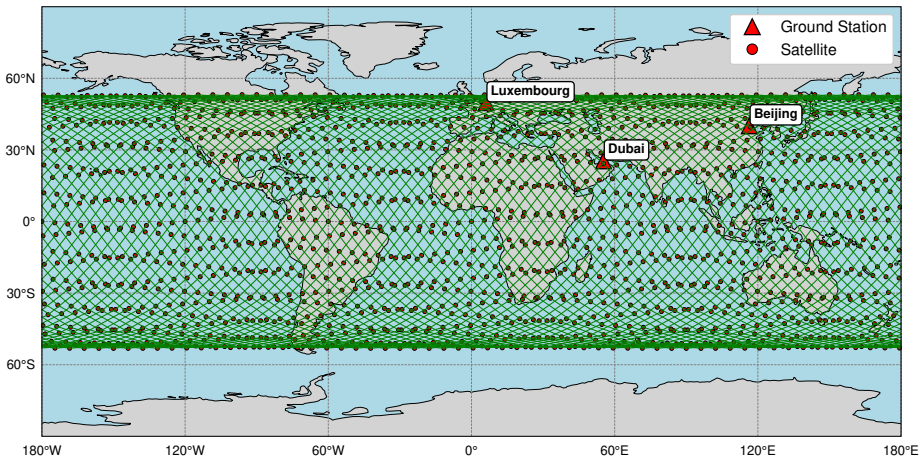

Figure 5.2 Network topology used in simulations.

distributional one and optimizing against CVaR, **PRIMAL-CVaR** provides a principled framework for building a robust routing policy that is explicitly sensitive to tail-end risks. This is critical for ensuring reliable performance in highly dynamic LEO networks. To summarize our proposed **PRIMAL** algorithm, we present its pseudo code in Algorithm 11 which integrates the two variants together for clarity. Note that we use a shared centralized replay buffer here by following the Centralized Training and Decentralized Execution (CTDE) paradigm during offline training. However, it can be easily extended to use private replay buffers via online federated learning, as shown in [29].

5.4 Experiment

5.4.1 Environmental Settings

To empirically validate our **PRIMAL** framework, we developed an asynchronous event-driven high-fidelity simulator using Python and PyTorch. Specifically, as shown in Fig. 5.2, we run simulations in an ultra-dense Walker-Delta LEO satellite with 22 satellites per orbit, and 72 evenly distributed orbits at the same altitude of

600 km in this Starlink-like constellation (1584 satellites). The inclination is $53°$ and the minimum elevation angle is $15°$. We generate packets from three ground stations representing three cities on Earth: Luxembourg, Dubai, and Beijing. All cities have equal probability to be chosen as the source or destination node of a generated packet. We update the satellite positions every 100 ms. We set stable link data rates at 1000 Mbps for GSLs and 50 Mbps for ISLs. The node and link buffer sizes are both 16 Mbits. We set the traffic packet length following the same setting as in [29], with 80% being normal packets (64.8 Kbits) and the remaining 20% being small packets (16.2 Kbits). In addition, we set the maximum TTL as $H = 64$. We run the simulation by a 30-second training or evaluation epoch with a packet traffic rate of $10,000$ packets/s, totaling $300,000$ packets per run according to the Poisson process. We run training iteration once every 1 ms, and report training performance metrics every 2 seconds (2K iterations).

For the neural network implementation, the actor (π_θ) and critics (Q^r_ϕ, $Q^c_{\psi_k}$) use a shared backbone, a two-layer MLP with 512 hidden units, to process observations. Each component has a separate MLP output head. For the **`PRIMAL-CVaR`** variant, the cost critic $Q^c_{\psi_k}$ is an IQN suggested in [143] with two layers and quantile parameters $N = N' = N^k = 64$. All algorithms use a mini-batch size of 1024, a replay buffer size of 300000, and discount factors $\gamma_r = 0.99$ and $\gamma_c = 0.97$.

For the cost and reward functions, we define the cost function as the normalized queuing delay $c_h = D^Q_h / D_{norm}$, where D^Q_h is the queuing delay experienced when packet p is forwarded over a link at hop h, and $D_{norm} = 100$ ms is a predefined constant for normalization. This cost directly quantifies the level of local congestion. The reward function is designed to minimize delay and maximize delivery rates by combining a dense progressive reward and a terminal reward B_p. The per-hop reward is defined as:

$$r_h = \frac{\tau}{D_{norm}} - c_h + \Delta d + B_p \tag{5.36}$$

where τ is the total action delay, and Δd provides a dense reward for geographic progress towards the destination, which measured by the difference of Great Circle Distance (GCD). A large terminal reward B_p is added at the final hop to prioritize successful packet delivery:

$$B_p = \begin{cases} 1 + L_p, & \text{if } p \text{ is delivered,} \\ -\frac{5\tau_p^{ttl}}{D_{norm}} - \sum_{j=0}^{h} \Delta d_{GCD}, & \text{if } p \text{ is dropped,} \\ 0, & \text{else,} \end{cases} \tag{5.37}$$

which encourages maximizing the packet delivery rate. For comparison, we use a hand-crafted reward

$$\bar{r}(o, a, o') = r(o, a, o') + \sum_{k=1}^{K} c_k(o, a, o') \tag{5.38}$$

as the reward function used by heuristic reward shaping approaches, which is equivalent to $\lambda_k = 1$ as all these components are well normalized. In addition, we set the minimum entropy as $\bar{\mathcal{H}} \approx 0.067$, which is a heuristic value considering the best action confidence to be 0.99 while the rest actions have the same probability of $\frac{1-0.99}{|\mathcal{A}|-1}$.

In all simulations, we set the threshold for the queuing delay cost as $D_{max}^{Q} = 10$ ms in total for each packet, and we compare the following algorithms:

- **`SPF`**: The Dijkstra's Shortest Path First (SPF) algorithm, assuming that routing table as precomputed based on the predictable orbital movements.
- **`MADQN`**: The multi-agent asynchronous DQN proposed in [29], using (5.38) as the effective reward function that relies on human prior knowledge.
- **`PRIMAL-Avg`** and **`PRIMAL-CVaR`**(ϵ): The two variants introduced in this paper, with a risk level $\epsilon \in [0, 1]$. Note that [36] uses a similar expectation based cost critic as in **`PRIMAL-Avg`**. However, [36] is not asynchronous and

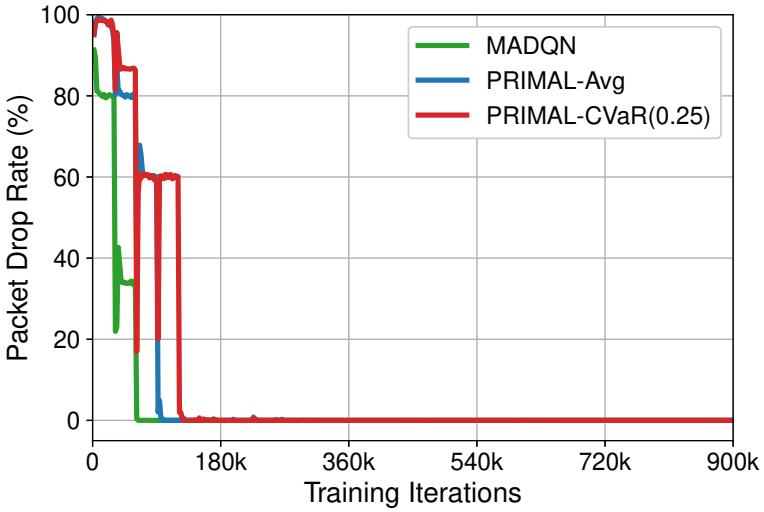

Figure 5.3 Average packet drop rate versus training epochs for RL algorithms

requires impractical synchronized joint-actions. While it can not work in our asynchronous simulator, we can treat **PRIMAL-Avg** as an asynchronous alternative for it.

5.4.2 Simulation Results

We now presents the empirical evaluation of our proposed PRIMAL algorithm against baseline methods. Figs.5.3-5.6 illustrate the training performance of the reinforcement learning agents. Specifically, Fig. 5.3 shows that all learning-based algorithms, **MADQN**, **PRIMAL-Avg**, and **PRIMAL-CVaR**, quickly learn to minimize the packet drop rate, achieving an almost-zero drop rate after approximately 150K training iterations. This indicates that all agents successfully learn the primary objective of delivering packets to their destinations. However, the algorithms show significant differences in their ability to manage network delay and constraints. As shown in Fig. 5.4, while all agents reduce the E2E packet delay over time, the **PRIMAL** variants achieve significantly better performance. **PRIMAL-CVaR** converges to the lowest average E2E delay of approximately 62 ms, followed

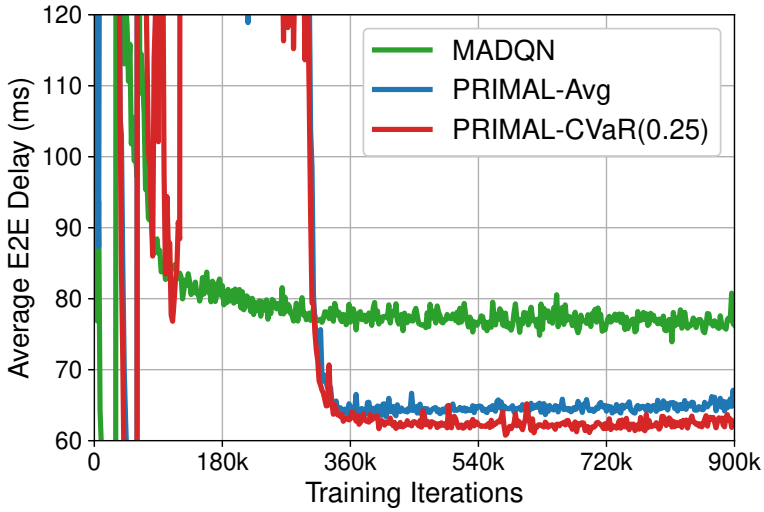

Figure 5.4 Average E2E delay versus training epochs for RL algorithms

by **PRIMAL-Avg** at around 66 ms, whereas **MADQN** stabilizes at a higher delay of about 77 ms.

The difference in performance is largely explained by how each algorithm handles the queuing delay constraint, which is a direct indicator for network congestion. Fig. 5.5 demonstrates that **MADQN** fails to respect the 10 ms queuing delay threshold, converging to a value consistently near 18 ms. In contrast, **PRIMAL-Avg**, which directly optimizes for the expected cost, successfully learns to keep the queuing delay at the 10 ms threshold. **PRIMAL-CVaR** goes one step further, reducing the average queuing delay to just 5 ms well below the threshold, by actively mitigating worst-case tail events. The unique capability of **PRIMAL-CVaR** is clearly validated in Fig. 5.6, which plots the $\mathrm{CVaR}_{0.25}$ of the queuing delay. We can find that only **PRIMAL-CVaR** successfully reduces the tail-end risk, bringing the $\mathrm{CVaR}_{0.25}$ of the queuing delay down to the 10 ms threshold, whereas both **MADQN** and **PRIMAL-Avg** exhibit a $\mathrm{CVaR}_{0.25}$ far exceeding this limit. This strongly confirms that **PRIMAL-CVaR** is highly risk-aware and is capable to effectively constrain worst-case congestion events.

The post-training evaluation results are summarized in Tables 5.1 and 5.2 and

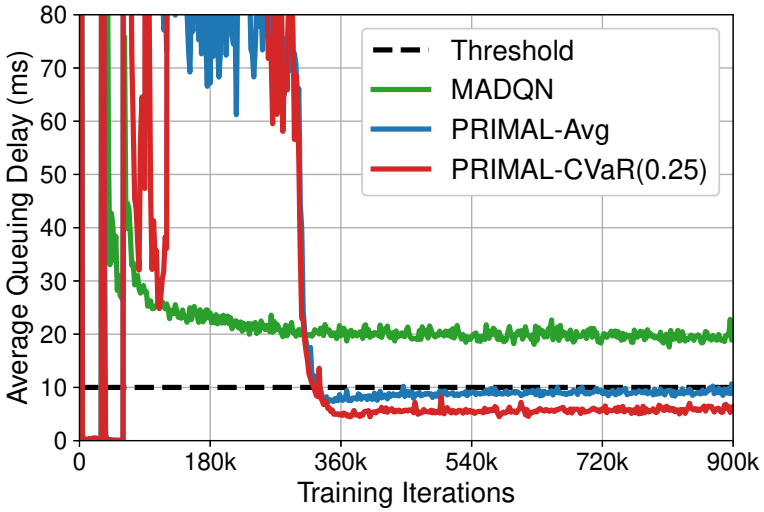

Figure 5.5 Average queuing delay versus training epochs for RL algorithms

detailed in Figs. 5.7 and 5.8, averaged over five test runs with different random seeds. The static baseline **SPF** performs poorly, suffering an 84.8% drop rate and high E2E delay variance. This shows its inability to handle dynamic congestion. In contrast, all learning-based methods achieve high throughput and near-zero drop rates. Among them, **PRIMAL-CVaR** delivers the best overall performance: the highest throughput (543.0 Mbps), the lowest E2E delay (61.5 ± 18.2 ms), and minimal queuing delay (4.8 ± 3.0 ms). **PRIMAL-Avg** also satisfies the average queuing delay constraint (8.9 ± 5.3 ms) and outperforms **MADQN**, which violates the constraint with 17.6 ± 10.1 ms due to its heuristic risk handling. To assess risk, we report $\mathrm{CVaR}_{0.25}$ of queuing delay: **MADQN** and **PRIMAL-Avg** incur 31.1 ms and 16.0 ms, while only **PRIMAL-CVaR** keeps it below the 10 ms threshold at 8.9 ms, indicating effective tail-risk mitigation. As a result, it also achieves the lowest violation magnitude, which outperforms all other methods.

Moreover, Figs. 5.7 and 5.9 provide a distributional view of the E2E and queuing delays. The results for SPF and MADQN show very high variance, indicating unpredictable performance. In contrast, the PRIMAL variants, particularly PRIMAL-CVaR, show much tighter delay distributions. In Fig. 5.9, while both the

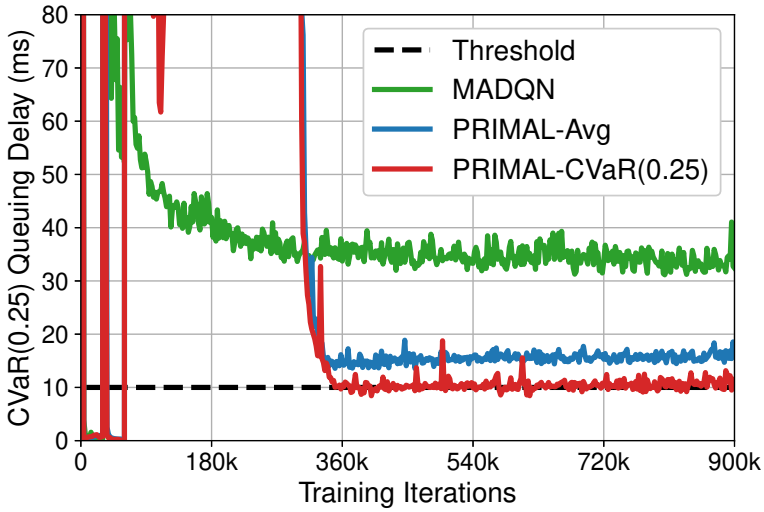

Figure 5.6 $\mathrm{CVaR}_{0.25}$ Queuing delays versus training epochs for RL algorithms

Algorithm	Throughput	Drop Rate	E2E Delay
SPF	27.0 Mbps	84.8%	62.0 ± 85.0 ms
MADQN	542.7 Mbps	**0.00%**	73.4 ± 20.4 ms
PRIMAL-Avg	542.9 Mbps	**0.00%**	64.6 ± 17.7 ms
PRIMAL-CVaR	**543.0** Mbps	**0.00%**	**61.5** ± **18.2** ms

Table 5.1 Routing performance comparison (Part 1: Core Metrics).

Algorithm	Queuing Delay		Load Balancing Constraint	
	mean ± std	$\mathrm{CVaR}_{0.25}$	Violation Rate[a]	Magnitude of violation[b]
SPF	**17.5** ± 80.2 ms	**70.1 ms**	85.7%	261.12 ± 176.60 ms
MADQN	**17.6** ± 10.1 ms	**31.1 ms**	75.5%	11.60 ± 8.10 ms
PRIMAL-Avg	**8.9** ± 5.3 ms	**16.0 ms**	38.6%	4.19 ± 3.35 ms
PRIMAL-CVaR	**4.8** ± 3.0 ms	**8.9 ms**	**5.8%**	**2.47** ± **2.38** ms

[a] Dropped packets are also included.
[b] Only counts for delivered packets.

Table 5.2 Routing performance comparison (Part 2: Delay & Constraint Metrics).

two PRIMAL variants keep the average queuing delay below the 10ms threshold, the distributions reveal a key difference in risk management. PRIMAL-Avg exhibits a wider distribution with a heavier tail, and its CVaR at a 25% risk level ($\mathrm{CVaR}_{0.25}$) significantly violates the threshold. This demonstrates PRIMAL-CVaR's ability to not only optimize for average performance but to also effectively mitigate

high-delay tail events, resulting in a more predictable and robust routing policy.

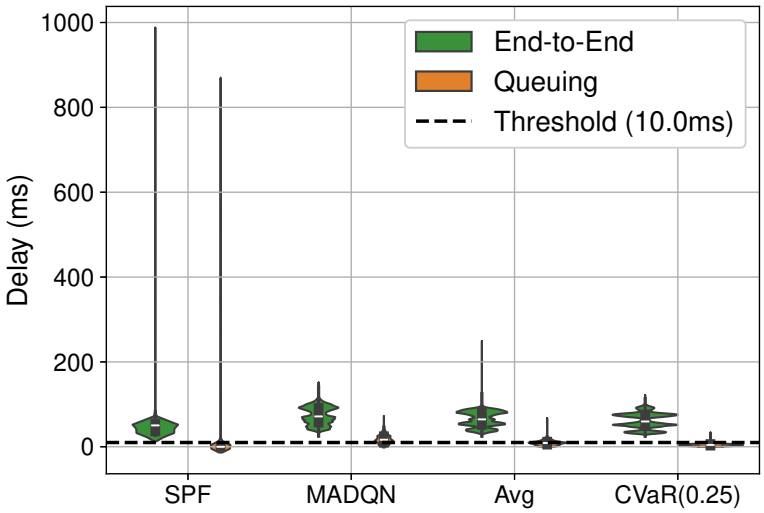

Figure 5.7 Evaluated delay distribution comparison for the competing algorithms

Fig. 5.8 breaks down the average end-to-end delay into its constituent parts, offering crucial insights into the operational differences between the routing strategies. From the delay compositions of the **PRIMAL** variants, a well-known (and almost common sense) principle is clearly validated: the fastest path is not always the geographically shortest one. For example, **PRIMAL-CVaR** accepts a minor increase in propagation delay over **PRIMAL-Avg** (only 1 ms higher), indicating it selects physically longer routes. However, what makes this result interesting is not the validation of the principle, but how our algorithm effectively realizes this well-know trade-off. With the minimal detour cost, **PRIMAL-CVaR** achieves a massive 46% reduction in queuing delay comparing to **PRIMAL-Avg**, leading to more effective and balanced network load. This highlights that risk-aware congestion avoidance is the most critical factor influencing performance, far outweighing the marginal cost of a slightly longer path. By making this intelligent risk-aware trade-off, **PRIMAL-CVaR** effectively bypasses network hotspots to achieve the lowest overall end-to-end delay with almost-zero packet dropping rate.

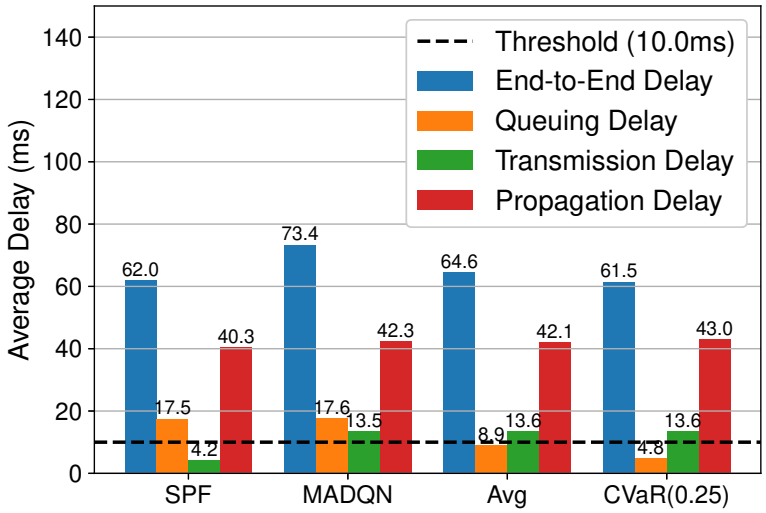

Figure 5.8 Evaluated delay components comparison for the competing algorithms

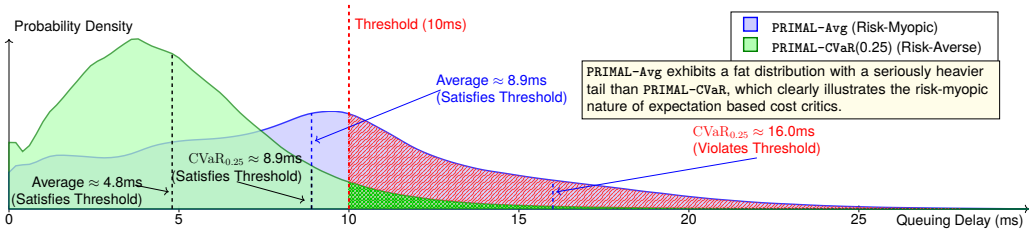

Figure 5.9 Learned policy's queuing delay distribution comparison for the proposed `PRIMAL` variants. Data were directly drawn from well-trained models.

5.5 Conclusion

In this chapter, we proposed **PRIMAL**, an asynchronous risk-aware multi-agent packet routing framework tailored for the dynamic decentralized nature of LEO satellite networks. Our event-driven design enables each satellite to act independently with their own pace, while risk-awareness is achieved through primal-dual learning using distributional RL to capture routing cost distributions and constrain tail risks via CVaR. Empirical results show that **PRIMAL** effectively avoids traffic hotspots by trading off slightly longer paths for improved load balancing and overall performance, demonstrating a principled approach to risk-aware routing

in highly dynamic networks.

6 Conclusion and Future Directions

This dissertation has addressed the critical challenge of designing intelligent and autonomous decision-making agents for next-generation communication systems. The evolution towards 6G is defined by an unprecedented increase in scale, decentralization, and dynamism, leading to a fundamental condition we have termed *partial observability at scale*. In environments such as massive MIMO systems and LEO mega-constellations, obtaining a complete and timely view of the true system state is often infeasible. This inherent uncertainty poses significant risks, as traditional control methods that optimize for average performance are often *risk-oblivious* or *risk-myopic*, failing to prevent high-impact tail-end events like severe QoS degradation or network congestion.

Recognizing this critical gap, this dissertation proposed a unified and principled framework for *risk-aware intelligence*. The central research question has been: *How can communication agents make robust and risk-aware decisions under large-scale partial observability?* To answer this, we moved beyond heuristic or average-case optimization to develop methodologies that explicitly model uncertainty and manage the risk of violating operational constraints. We instantiated this framework through two complementary paradigms: model-based risk-aware planning and model-free risk-aware multi-agent reinforcement learning.

6.1 Key Achievements

The primary contribution of this work is a comprehensive set of tools and methodologies that enable robust decision-making under uncertainty. The key achievements are summarized below.

6.1.1 A Principled Framework for Risk-Aware Decision-Making

We established a formal paradigm that connects an agent's partial observations to its *belief* over the hidden system state, and in turn, connects this belief uncertainty to the *risk* of violating performance constraints. By quantifying risk as a direct function of state uncertainty, this approach provides a principled foundation for designing robust risk-aware intelligent agents in large-scale partially observable communication systems.

6.1.2 Model-Based Risk-Aware Planning for Massive MIMO

In Part I, we developed a model-based methodology for scenarios where system dynamics can be effectively learned and simulated. This was demonstrated in the context of antenna selection in massive MIMO systems with incomplete CSI.

Risk-Aware MCTS for Antenna Selection

We introduced the Joint Channel Prediction and Antenna Selection (JCPAS) framework, centered around a novel *Risk-Aware Monte-Carlo Tree Search (RA-MCTS)* planner. Unlike conventional approaches that rely on a single-point estimate of the channel, RA-MCTS leverages a predictive *"world model"* to forecast a *belief distribution* over future channel states. By planning over this distribution, the agent can select actions that explicitly reduce the probability of future QoS violations, thereby managing risk in a principled manner.

An Advanced Spatio-Temporal World Model

The efficacy of any model-based planner is dependent on the accuracy of its world model. To this end, we developed an advanced spatio-temporal predictive model featuring a novel *Crossover Attention (XOA)* mechanism. By modifying the standard Transformer architecture to explicitly and simultaneously capture both spatial and temporal correlations, our XOA-enhanced model provides more accurate predictions from history, directly enabling more robust and informed decision-making by the model-based planner.

6.1.3 Model-Free Risk-Aware MARL for LEO Constellations

In Part II, we addressed decentralized systems where learning an accurate *"world model"* is intractable. We developed a model-free methodology demonstrated on the challenging problem of asynchronous packet routing in LEO satellite mega-constellations.

Asynchronous Event-Driven Multi-Agent Learning

We proposed the *PRIMAL* framework, which is uniquely designed for the asynchronous, event-driven nature of LEO networks. By modeling the problem as a collection of independent semi-Markov decision processes, PRIMAL allows each satellite to make routing decisions based on its local event timeline, eliminating the unrealistic and inefficient action-synchronization assumptions common in prior MARL-based routing algorithms.

Distributional Primal-Dual Learning for Tail-Risk Control

To manage the risks arising from uncoordinated decentralized actions under partially observability, PRIMAL employs a principled *distributional primal-dual learning* method. Instead of learning only the expected QoS metrics, agents learn the full conditional probability distribution of routing outcomes. This enables

the direct optimization and constraint of the *CVaR*, a coherent risk measure that quantifies worst-case outcomes. By constraining the CVaR, PRIMAL effectively mitigates tail-end risks such as severe latency spikes and network congestion, ensuring robust performance where average-case optimization would fail.

6.2 Limitations and Future Research Directions

While this dissertation provides a robust foundation for risk-aware intelligence, it also opens up several exciting potential improvements for future research. The limitations of the current work and potential future directions are intertwined.

6.2.1 Limitations of the Current Work

The proposed paradigms, while effective, have the following limitations.

- *Model-Based Paradigm:* The performance of the RA-MCTS planner is fundamentally bounded by the accuracy of its predictive world model. In highly non-stationary environments, model mismatch could degrade performance. Furthermore, the computational complexity of MCTS, while managed, can still be a concern for real-time applications with extremely tight latency budgets.

- *Model-Free Paradigm:* The PRIMAL framework relies on independent learners for scalability. While effective, this may lead to less coordinated behavior compared to centralized training paradigms in certain scenarios. Additionally, the sample efficiency of model-free RL remains a general challenge, potentially requiring significant interaction with the environment to converge to an optimal policy.

6.2.2 Future Directions

Building upon the contributions and limitations of this work, we identify the following promising research directions.

Hybrid Model-Based and Model-Free Approaches

A powerful direction is to bridge the gap between the two paradigms developed in this thesis. A learned world model, even an imperfect one, could be used to generate synthetic experiences to augment the replay buffer of a model-free agent like PRIMAL. This approach, inspired by architectures like Dyna [16], could dramatically improve sample efficiency and accelerate learning, combining the planning capabilities of model-based methods with the robustness of model-free learning.

Foundation World Models for Communication Networks

A truly transformative direction would be the development of a *foundation world model for communication systems*. Analogous to how Large Language Models (LLMs) are pre-trained on vast text corpora, one could pre-train a massive Transformer-based model on diverse datasets of network dynamics from simulations and real-world deployments. Such a model could serve as a powerful, general-purpose world model, capable of being fine-tuned for a wide array of downstream tasks, from resource allocation to routing, significantly accelerating the development of model-based risk-aware intelligence. This represents a paradigm shift from building bespoke models for each problem to leveraging a single, powerful, pre-trained understanding of network physics and dynamics.

Provably Efficient Risk-Awareness

One critical direction for the future research is providing formal, rigorous, and mathematical guarantees on performance and safety, leading to *provable risk-*

awareness. The primal-dual framework used in this dissertation offers a direct path toward this goal. For constrained optimization problems, the concept of the *duality gap* is central, i.e.,the difference between the solutions of the primal and dual problems. While a zero duality gap (strong duality) provides the strongest form of verification, it is often not guaranteed in decentralized multi-agent settings like the partially observable MARL problems studied here. Therefore, a significant open research problem is to analyze, quantify, and bound this duality gap for risk-aware MARL. Deriving such bounds would provide a formal certificate on the degree of sub-optimality and constraint satisfaction of the learned policy, making the agent's behavior not just effective, but mathematically verifiable and admissible.

Emergent Communication for Scalable Coordination

While the independent learning paradigm in `PRIMAL` ensures scalability, coordination among agents can be suboptimal as they have no direct mechanism to share their intentions. A fascinating future direction is to enable *emergent communication*, where agents learn not only how to act but also how to communicate. In this paradigm, agents could learn to decide when and how to send concise low-bandwidth messages to their neighbors to signal their intentions or share critical local state information (e.g., impending congestion). The crucial aspect is that the communication protocol is not pre-designed; rather, its meaning *emerges* as agents discover which messages are most effective for maximizing their collective long-term objectives. This would allow for a more dynamic and sophisticated form of decentralized coordination, mitigating potential conflicts and improving global performance.

References

[1] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato. "A Survey on Space-Air-Ground-Sea Integrated Network Security in 6G". In: *IEEE Commun. Surv. Tutorials* 24.1 (2022), pp. 53–87.

[2] S. Mahboob and L. Liu. "Revolutionizing Future Connectivity: A Contemporary Survey on AI-Empowered Satellite-Based Non-Terrestrial Networks in 6G". In: *IEEE Commun. Surv. Tutorials* 26.2 (2024), pp. 1279–1321.

[3] H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas, and S. Chatzinotas. "A Survey on Nongeostationary Satellite Systems: The Communication Perspective". In: *IEEE Commun. Surv. Tutorials* 25.1 (2023), pp. 101–132.

[4] S. Elhoushy, M. Ibrahim, and W. Hamouda. "Cell-Free Massive MIMO: A Survey". In: *IEEE Commun. Surv. Tutorials* 24.1 (2022), pp. 492–523.

[5] L. Sanguinetti, E. Björnson, and J. Hoydis. "Toward massive MIMO 2.0: Understanding spatial correlation, interference suppression, and pilot contamination". In: *IEEE Transactions on Communications* 68.1 (2019), pp. 232–257.

[6] F. Tang, B. Mao, Y. Kawamoto, and N. Kato. "Survey on Machine Learning for Intelligent End-to-End Communication Toward 6G: From Network Access, Routing to Traffic Control and Streaming Adaption". In: *IEEE Commun. Surv. Tutorials* 23.3 (2021), pp. 1578–1598.

[7] O. Kodheli et al. “Satellite Communications in the New Space Era: A Survey and Future Challenges”. In: *IEEE Commun. Surv. Tutorials* 23.1 (2021), pp. 70–109.

[8] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang. “An overview of massive MIMO: Benefits and challenges”. In: *IEEE J. Sel. Top. Signal Process.* 8.5 (2014), pp. 742–758.

[9] Z. Wang, J. Zhang, H. Du, D. Niyato, S. Cui, B. Ai, M. Debbah, K. B. Letaief, and H. V. Poor. “A Tutorial on Extremely Large-Scale MIMO for 6G: Fundamentals, Signal Processing, and Applications”. In: *IEEE Commun. Surv. Tutorials* 26.3 (2024), pp. 1560–1605.

[10] S. Cakaj. “The parameters comparison of the Starlink LEO satellites constellation for different orbital shells”. In: *Front. Commun. Netw.* 2 (2021), p. 643095.

[11] T. L. Marzetta, E. G. Larsson, and H. Yang. *Fundamentals of massive MIMO*. Cambridge University Press, 2016.

[12] L. Lu, G. Y. Li, A. L. Swindlehurst, A. E. Ashikhmin, and R. Zhang. “An Overview of Massive MIMO: Benefits and Challenges”. In: *IEEE J. Sel. Top. Signal Process.* 8.5 (2014), pp. 742–758.

[13] L. You, K. Li, J. Wang, X. Gao, X. Xia, and B. E. Ottersten. “LEO Satellite Communications with Massive MIMO”. In: *Proc. IEEE ICC*. 2020, pp. 1–6.

[14] Z. M. Bakhsh, Y. Omid, G. Chen, F. Kayhan, Y. Ma, and R. Tafazolli. “Multi-Satellite MIMO Systems for Direct Satellite-to-Device Communications: A Survey”. In: *IEEE Commun. Surv. Tutorials* 27.3 (2025), pp. 1536–1564.

[15] D. S. Lakew, U. Sa’ad, N. Dao, W. Na, and S. Cho. “Routing in Flying Ad Hoc Networks: A Comprehensive Survey”. In: *IEEE Commun. Surv. Tutorials* 22.2 (2020), pp. 1071–1120.

[16] F. Luo, T. Xu, H. Lai, X. Chen, W. Zhang, and Y. Yu. "A survey on model-based reinforcement learning". In: *Sci. China Inf. Sci.* 67.2 (2024).

[17] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll. "A Review of Safe Reinforcement Learning: Methods, Theories, and Applications". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 46.12 (2024), pp. 11216–11235.

[18] Y. Yang, X. He, J. Lee, D. He, and Y. Li. "Collaborative Deep Reinforcement Learning in 6G Integrated Satellite-Terrestrial Networks: Paradigm, Solutions, and Trends". In: *IEEE Commun. Mag.* 63.1 (2025), pp. 188–195.

[19] S. Sharifi, S. Shahbazpanahi, and M. Dong. "A POMDP-based antenna selection for massive MIMO communication". In: *IEEE Trans. Commun.* 70.3 (2021), pp. 2025–2041.

[20] J. Xu and D. Yang. "Energy-efficient resource allocation for D2D communication underlaying cellular networks with incomplete CSI". In: *Comput. Networks* 251 (2024), p. 110664.

[21] S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro. "Constrained reinforcement learning has zero duality gap". In: *Adv. Neural Inf. Process. Syst.* Vol. 32. 2019.

[22] Y. Lv, C. Xing, N. Xu, X. Han, and F. Wang. "Research of adaptive routing scheme for LEO network". In: *Proc. IEEE ICCC*. 2019, pp. 987–992.

[23] T. K. Vu, M. Bennis, M. Debbah, M. Latva-aho, and C. S. Hong. "Ultra-Reliable Communication in 5G mmWave Networks: A Risk-Sensitive Approach". In: *IEEE Commun. Lett.* 22.4 (2018), pp. 708–711.

[24] Q. Zhang, S. Leng, X. Ma, Q. Liu, X. Wang, B. Liang, Y. Liu, and J. Yang. "CVaR-Constrained Policy Optimization for Safe Reinforcement Learning". In: *IEEE Trans. Neural Networks Learn. Syst.* 36.1 (2025), pp. 830–841.

[25] J. Chen, S. Chen, Y. Qi, and S. Fu. “Intelligent massive MIMO antenna selection using Monte Carlo tree search”. In: *IEEE Trans. Sig. Proc.* 67.20 (2019), pp. 5380–5390.

[26] C. F. Hayes, M. Reymond, D. M. Roijers, E. Howley, and P. Mannion. “Distributional monte carlo tree search for risk-aware and multi-objective reinforcement learning”. In: (2021), pp. 1530–1532.

[27] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, et al. “A tutorial on Thompson sampling”. In: *Found. Trends Mach. Learn.* 11.1 (2018), pp. 1–96.

[28] S. Li, Q. Wu, R. Wang, and R. Lu. “Toward Networking and Routing in 6G Satellite-Terrestrial Integrated Networks: Current Issues and a Potential Solution”. In: *IEEE Commun. Mag.* 63.3 (2025), pp. 92–98.

[29] F. Lozano-Cuadra, B. Soret, I. Leyva-Mayorga, and P. Popovski. “Continual deep reinforcement learning for decentralized satellite routing”. In: *IEEE Trans. Commun.* (2025).

[30] M. Gharavi-Alkhansari and A. B. Gershman. “Fast antenna subset selection in MIMO systems”. In: *IEEE Trans. Sig. Proc.* 52.2 (2004), pp. 339–347.

[31] Z. Kuai and S. Wang. “Thompson sampling-based antenna selection with partial CSI for TDD massive MIMO systems”. In: *IEEE Trans. Commun.* 68.12 (2020), pp. 7533–7546.

[32] K. He, T. X. Vu, D. T. Hoang, D. N. Nguyen, S. Chatzinotas, and B. Ottersten. “Risk-Aware Antenna Selection for Multiuser Massive MIMO under Incomplete CSI”. In: *IEEE Trans. Wirel. Commun.* 23.9 (2024), pp. 11001–11014.

[33] S. Sharifi. “A POMDP framework for antenna selection and user scheduling in multi-user massive MIMO systems”. PhD thesis. University of Ontario Institute of Technology, 2022.

[34] S. Sharifi and S. Shahbazpanahi. "A POMDP-Based Approach to Joint Antenna Selection and User Scheduling for Multi-User Massive MIMO Communication". In: *IEEE Trans. Commun.* 71.3 (2023), pp. 1691–1706.

[35] W. Wei, L. Fu, H. Gu, X. Lu, L. Liu, S. Mumtaz, and M. Guizani. "Iris: Toward Intelligent Reliable Routing for Software-Defined Satellite Networks". In: *IEEE Trans. Commun.* 73.1 (2025), pp. 454–468.

[36] Y. Lyu, H. Hu, R. Fan, Z. Liu, J. An, and S. Mao. "Dynamic Routing for Integrated Satellite-Terrestrial Networks: A Constrained Multi-Agent Reinforcement Learning Approach". In: *IEEE J. Sel. Areas Commun.* 42.5 (2024), pp. 1204–1218.

[37] K. He, T. X. Vu, L. He, L. Fan, S. Chatzinotas, and B. Ottersten. "Asynchronous Risk-Aware Multi-Agent Packet Routing for Ultra-Dense LEO Satellite Networks". In: *IEEE International Conference on Computer Communications (INFOCOM).* Submitted for review. 2026.

[38] K. He, T. X. Vu, L. Fan, S. Chatzinotas, and B. Ottersten. "Spatio-Temporal Predictive Learning Using Crossover Attention for Communications and Networking Applications". In: *IEEE Trans. Mach. Learn. Commun. Netw.* (2025).

[39] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* Second. The MIT Press, 2018.

[40] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. "A survey of Monte Carlo tree search methods". In: *IEEE Trans. Comput. Intell. AI Games* 4.1 (2012), pp. 1–43.

[41] H. Van Hasselt, A. Guez, and D. Silver. "Deep reinforcement learning with double q-learning". In: *Proc. AAAI.* Vol. 30. 1. 2016.

[42] S. Levine. “Reinforcement learning and control as probabilistic inference: Tutorial and review”. In: *arXiv preprint arXiv:1805.00909* (2018).

[43] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. In: *Proc. ICML*. 2018, pp. 1861–1870.

[44] H. Zhou, T. Wei, Z. Lin, J. Li, J. Xing, Y. Shi, L. Shen, C. Yu, and D. Ye. “Revisiting Discrete Soft Actor-Critic”. In: *Trans. Mach. Learn. Res.* 2024 (2024).

[45] K. Zhang, Z. Yang, and T. Başar. “Multi-agent reinforcement learning: A selective overview of theories and algorithms”. In: *Handbook of reinforcement learning and control* (2021), pp. 321–384.

[46] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson. “Is independent learning all you need in the starcraft multi-agent challenge?” In: *arXiv preprint arXiv:2011.09533* (2020).

[47] Y. Liang, H. Wu, and H. Wang. “ASM-PPO: Asynchronous and scalable multi-agent PPO for cooperative charging”. In: *Proc. AAMAS*. 2022, pp. 798–806.

[48] Y. Xiao, W. Tan, and C. Amato. “Asynchronous Actor-Critic for Multi-Agent Reinforcement Learning”. In: *Adv. Neural Inf. Process. Syst.* 2022.

[49] C. Yu et al. “Asynchronous Multi-Agent Reinforcement Learning for Efficient Real-Time Multi-Robot Cooperative Exploration”. In: *Proc. AAMAS*. 2023, pp. 1107–1115.

[50] C. Le, T. X. Vu, and S. Chatzinotas. “Cooperative UAVs with Asynchronous Multi-agent Learning for Remote Data Collection”. In: *Proc. Globecom Wks.* 2024.

[51] J. Wang, X. Zhang, X. Shi, and J. Song. "Higher spectral efficiency for mmWave MIMO: Enabling techniques and precoder designs". In: *IEEE Commun. Mag.* 59.4 (2021), pp. 116–122.

[52] W.-B. Sun, W.-X. Meng, J.-C. Guo, and C. Li. "Multiuser MIMO Opportunistic Beamforming Communications: State-of-the-Art and Perspectives". In: *IEEE Wirel. Commun.* 29.3 (2022), pp. 95–101.

[53] A. F. Molisch, V. V. Ratnam, S. Han, Z. Li, S. L. H. Nguyen, L. Li, and K. Haneda. "Hybrid Beamforming for Massive MIMO: A Survey". In: *IEEE Commun. Mag.* 55.9 (2017), pp. 134–141.

[54] Y. Gao, H. Vinck, and T. Kaiser. "Massive MIMO antenna selection: Switching architectures, capacity bounds, and optimal antenna selection algorithms". In: *IEEE Trans. Sig. Proc.* 66.5 (2017), pp. 1346–1360.

[55] A. Gorokhov. "Antenna selection algorithms for MEA transmission systems". In: *Proc. ICASSP*. Vol. 3. 2002, pp. III–2857.

[56] P.-H. Lin and S.-H. Tsai. "Performance analysis and algorithm designs for transmit antenna selection in linearly precoded multiuser MIMO systems". In: *IEEE Trans. Veh. Technol.* 61.4 (2012), pp. 1698–1708.

[57] S. Zhang, S. Zhang, F. Gao, J. Ma, and O. A. Dobre. "Deep learning optimized sparse antenna activation for reconfigurable intelligent surface assisted communication". In: *IEEE Trans. Commun.* 69.10 (2021), pp. 6691–6705.

[58] T. X. Vu, S. Chatzinotas, V.-D. Nguyen, D. T. Hoang, D. N. Nguyen, M. Di Renzo, and B. Ottersten. "Machine learning-enabled joint antenna selection and precoding design: From offline complexity to online performance". In: *IEEE Trans. Wirel. Commun.* 20.6 (2021), pp. 3710–3722.

[59] W. Yu, T. Wang, and S. Wang. “Multi-label learning based antenna selection in massive MIMO systems”. In: *IEEE Trans. Veh. Technol.* 70.7 (2021), pp. 7255–7260.

[60] Z. Liu, Y. Yang, F. Gao, T. Zhou, and H. Ma. “Deep Unsupervised Learning for Joint Antenna Selection and Hybrid Beamforming”. In: *IEEE Trans. Commun.* 70.3 (2022), pp. 1697–1710.

[61] O. Raeesi, A. Gokceoglu, Y. Zou, E. Björnson, and M. Valkama. “Performance analysis of multi-user massive MIMO downlink under channel non-reciprocity and imperfect CSI”. In: *IEEE Trans. Commun.* 66.6 (2018), pp. 2456–2471.

[62] X. Gao, O. Edfors, F. Tufvesson, and E. G. Larsson. “Massive MIMO in real propagation environments: Do all antennas contribute equally?” In: *IEEE Trans. Commun.* 63.11 (2015), pp. 3917–3928.

[63] K. He, T. X. Vu, S. Chatzinotas, and B. Ottersten. “Learning-Based Joint Channel Prediction and Antenna Selection for Massive MIMO with Partial CSI”. In: *IEEE GLOBECOM Workshops*. 2022, pp. 178–183.

[64] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I. Wong, V. Öwall, O. Edfors, and F. Tufvesson. “A flexible 100-antenna testbed for massive MIMO”. In: *IEEE GLOBECOM Workshops*. 2014, pp. 287–293.

[65] H. S. Wang and P.-C. Chang. “On verifying the first-order Markovian assumption for a Rayleigh fading channel model”. In: *IEEE Trans. Veh. Technol.* 45.2 (1996), pp. 353–357.

[66] C. C. Tan and N. C. Beaulieu. “On first-order Markov modeling for the Rayleigh fading channel”. In: *IEEE Trans. Commun.* 48.12 (2000), pp. 2032–2040.

[67] G. J. Byers and F. Takawira. “Spatially and temporally correlated MIMO channels: Modeling and capacity analysis”. In: *IEEE Trans. Veh. Technol.* 53.3 (2004), pp. 634–643.

[68] C. Wu, X. Yi, Y. Zhu, W. Wang, L. You, and X. Gao. “Channel prediction in high-mobility massive MIMO: From spatio-temporal autoregression to deep learning”. In: *IEEE J. Sel. Areas Commun.* 39.7 (2021), pp. 1915–1930.

[69] T. Zhou, H. Zhang, B. Ai, C. Xue, and L. Liu. “Deep-Learning Based Spatial-Temporal Channel Prediction for Smart High-Speed Railway Communication Networks”. In: *IEEE Trans. Wirel. Commun.* (2022).

[70] G. Liu, Z. Hu, L. Wang, J. Xue, H. Yin, and D. Gesbert. “Spatio-Temporal Neural Network for Channel Prediction in Massive MIMO-OFDM Systems”. In: *IEEE Trans. Commun.* 70.12 (2022), pp. 8003–8016.

[71] N. B. Khalifa, M. Assaad, and M. Debbah. “Risk-Sensitive Reinforcement Learning for URLLC Traffic in Wireless Networks”. In: *Proc. IEEE WCNC*. 2019, pp. 1–7.

[72] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong. “Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach”. In: *IEEE Trans. Wirel. Commun.* 20.7 (2021), pp. 4585–4600.

[73] L.-C. Lan, T.-R. W. Wu, I.-C. W. Wu, and C.-J. Hsieh. “Learning to stop: Dynamic simulation Monte-Carlo tree search”. In: *AAAI Conference on Artificial Intelligence*. Vol. 35. 1. 2021, pp. 259–267.

[74] D. P. Kingma and P. Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *Proc. NeurIPS*. Vol. 31. 2018.

[75] A. Abdelhamed, M. A. Brubaker, and M. S. Brown. “Noise flow: Noise modeling with conditional normalizing flows”. In: *Proc. ICCV*. 2019, pp. 3165–3173.

[76] L. Dinh, D. Krueger, and Y. Bengio. “NICE: Non-linear independent components estimation”. In: *arXiv preprint arXiv:1410.8516* (2014).

[77] K. He, L. He, L. Fan, Y. Deng, G. K. Karagiannidis, and A. Nallanathan. “Learning-based signal detection for MIMO systems with unknown noise statistics”. In: *IEEE Trans. Commun.* 69.5 (2021), pp. 3025–3038.

[78] M. Phan, Y. Abbasi Yadkori, and J. Domke. “Thompson sampling and approximate inference”. In: *Proc. NeurIPS*. Vol. 32. 2019.

[79] D. Eckles and M. Kaptein. “Thompson sampling with the online bootstrap”. In: *arXiv preprint arXiv:1410.4009* (2014).

[80] K. He and J. Sun. “Convolutional neural networks at constrained time cost”. In: *Proc. CVPR*. 2015, pp. 5353–5360.

[81] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran. “Multiuser MIMO achievable rates with downlink training and channel state feedback”. In: *IEEE Trans. Inf. Theory* 56.6 (2010), pp. 2845–2866.

[82] E. Zöchmann, S. Schwarz, and M. Rupp. “Comparing antenna selection and hybrid precoding for millimeter wave wireless communications”. In: *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*. 2016, pp. 1–5.

[83] C. Wu, X. Yi, Y. Zhu, W. Wang, L. You, and X. Gao. “Channel Prediction in High-Mobility Massive MIMO: From Spatio-Temporal Autoregression to Deep Learning”. In: *IEEE J. Sel. Areas Commun.* 39.7 (2021), pp. 1915–1930.

[84] M. K. Shehzad, L. Rose, S. Wesemann, and M. Assaad. “ML-Based Massive MIMO Channel Prediction: Does It Work on Real-World Data?” In: *IEEE Wirel. Commun. Lett.* 11.4 (2022), pp. 811–815.

[85] S. Mehrizi and S. Chatzinotas. “Network Traffic Modeling and Prediction Using Graph Gaussian Processes”. In: *IEEE Access* 10 (2022), pp. 132644–132655.

[86] D. Cai, P. Fan, Q. Zou, Y. Xu, Z. Ding, and Z. Liu. "Active device detection and performance analysis of massive non-orthogonal transmissions in cellular internet of things". In: *Science China information sciences* 65.8 (2022), p. 182301.

[87] Q. Liu, J. Li, and Z. Lu. "ST-Tran: Spatial-Temporal Transformer for Cellular Traffic Prediction". In: *IEEE Commun. Lett.* 25.10 (2021), pp. 3325–3329.

[88] X. Wang, Z. Wang, K. Yang, Z. Song, C. Bian, J. Feng, and C. Deng. "A Survey on Deep Learning for Cellular Traffic Prediction". In: *Intelligent Computing* 3 (2024), p. 0054.

[89] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann. "LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting". In: *Proc. NeurIPS*. 2023.

[90] F. Chiariotti, M. Drago, P. Testolina, M. Lecci, A. Zanella, and M. Zorzi. "Temporal Characterization and Prediction of VR Traffic: A Network Slicing Use Case". In: *IEEE Trans. Mob. Comput.* 23.5 (2024), pp. 3890–3908.

[91] S. Siami-Namini, N. Tavakoli, and A. S. Namin. "A comparison of ARIMA and LSTM in forecasting time series". In: *Proc. ICMLA*. Ieee. 2018, pp. 1394–1401.

[92] C. Zhang, H. Zhang, D. Yuan, and M. Zhang. "Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks". In: *IEEE Commun. Lett.* 22.8 (2018), pp. 1656–1659.

[93] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui. "Spatio-Temporal Wireless Traffic Prediction With Recurrent Neural Network". In: *IEEE Wirel. Commun. Lett.* 7.4 (2018), pp. 554–557.

[94] C. Tan, Z. Gao, L. Wu, Y. Xu, J. Xia, S. Li, and S. Z. Li. "Temporal Attention Unit: Towards Efficient Spatiotemporal Predictive Learning". In: *Proc. CVPR 2023*. IEEE, 2023, pp. 18770–18782.

[95] G. Liu, Z. Hu, L. Wang, J. Xue, H. Yin, and D. Gesbert. “Spatio-Temporal Neural Network for Channel Prediction in Massive MIMO-OFDM Systems”. In: *IEEE Trans. Commun.* 70.12 (2022), pp. 8003–8016.

[96] T. Zhou, H. Zhang, B. Ai, C. Xue, and L. Liu. “Deep-Learning-Based Spatial-Temporal Channel Prediction for Smart High-Speed Railway Communication Networks”. In: *IEEE Trans. Wirel. Commun.* 21.7 (2022), pp. 5333–5345.

[97] F. Sun, P. Wang, J. Zhao, N. Xu, J. Zeng, J. Tao, K. Song, C. Deng, J. C. S. Lui, and X. Guan. “Mobile Data Traffic Prediction by Exploiting Time-Evolving User Mobility Patterns”. In: *IEEE Trans. Mob. Comput.* 21.12 (2022), pp. 4456–4470.

[98] T. Qi, G. Li, L. Chen, and Y. Xue. “ADGCN: An Asynchronous Dilation Graph Convolutional Network for Traffic Flow Prediction”. In: *IEEE Internet Things J.* 9.5 (2022), pp. 4001–4014.

[99] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li. “Long sequence time-series forecasting with deep learning: A survey”. In: *Information Fusion* 97 (2023), p. 101819.

[100] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. “Transformers in Time Series: A Survey”. In: (2023), pp. 6778–6786.

[101] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong. “Spatial-temporal transformer networks for traffic flow forecasting”. In: *arXiv preprint arXiv:2001.02908* (2020).

[102] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Proc. NIPS*. 2015, pp. 802–810.

[103] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai. "Accurate Channel Prediction Based on Transformer: Making Mobility Negligible". In: *IEEE J. Sel. Areas Commun.* 40.9 (2022), pp. 2717–2732.

[104] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang, and Z. Wang. "Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach". In: *IEEE Trans. Ind. Informatics* 19.4 (2023), pp. 5722–5731.

[105] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath. "An Attentive Survey of Attention Models". In: *ACM Trans. Intell. Syst. Technol.* 12.5 (2021), 53:1–53:32.

[106] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool. "Transformers in time-series analysis: A tutorial". In: *Circuits, Systems, and Signal Processing* 42.12 (2023), pp. 7433–7466.

[107] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar. "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting". In: *International conference on learning representations*. 2021.

[108] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. "Informer: Beyond efficient transformer for long sequence time-series forecasting". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 12. 2021, pp. 11106–11115.

[109] N. Kitaev, Ł. Kaiser, and A. Levskaya. "Reformer: The efficient transformer". In: *arXiv preprint arXiv:2001.04451* (2020).

[110] A. Das, W. Kong, R. Sen, and Y. Zhou. "A decoder-only foundation model for time-series forecasting". In: *arXiv preprint arXiv:2310.10688* (2023).

[111] S. Zheng, C. Shen, and X. Chen. "Design and analysis of uplink and downlink communications for federated learning". In: *IEEE Journal on Selected Areas in Communications* 39.7 (2020), pp. 2150–2167.

[112] R. Kumar, J. Mendes-Moreira, and J. Chandra. "Spatio-temporal parallel transformer based model for traffic prediction". In: *ACM Transactions on Knowledge Discovery from Data* 18.9 (2024), pp. 1–25.

[113] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri. "A multi-source dataset of urban life in the city of Milan and the Province of Trentino". In: *Scientific Data* 2.1 (2015), pp. 1–15.

[114] D. Bahdanau, K. Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *Proc. ICLR*. 2015.

[115] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[116] T. Wang, A. Roberts, D. Hesslow, T. Le Scao, H. W. Chung, I. Beltagy, J. Launay, and C. Raffel. "What language model architecture and pretraining objective works best for zero-shot generalization?" In: *International Conference on Machine Learning*. PMLR. 2022, pp. 22964–22984.

[117] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li, et al. "Large models for time series and spatio-temporal data: A survey and outlook". In: *arXiv preprint arXiv:2310.10196* (2023).

[118] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[119] Y. Li, R. Yu, C. Shahabi, and Y. Liu. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: (2018).

[120] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang. "Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting". In: *Proc. NIPS*. 2020.

[121] B. Yu, H. Yin, and Z. Zhu. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting". In: *Proc. IJCAI*. 2018, pp. 3634–3640.

[122] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, Z. Xiong, A. Jamalipour, and D. I. Kim. "Generative AI Agents With Large Language Model for Satellite Networks via a Mixture of Experts Transmission". In: *IEEE J. Sel. Areas Commun.* 42.12 (2024), pp. 3581–3596.

[123] Z. Han, C. Xu, G. Zhao, S. Wang, K. Cheng, and S. Yu. "Time-Varying Topology Model for Dynamic Routing in LEO Satellite Constellation Networks". In: *IEEE Trans. Veh. Technol.* 72.3 (2023), pp. 3440–3454.

[124] G. Zheng, N. Wang, R. Tafazolli, X. Wei, and J. Yang. "Virtual Data-Plane Addressing for SDN-based Space and Terrestrial Network Integration". In: *Proc. IEEE HPSR*. 2021, pp. 1–6.

[125] E. Ekici, I. F. Akyildiz, and M. D. Bender. "A distributed routing algorithm for datagram traffic in LEO satellite networks". In: *IEEE/ACM Trans. Netw.* 9.2 (2002), pp. 137–147.

[126] S. Li, Q. Wu, and R. Wang. "Dynamic Discrete Topology Design and Routing for Satellite-Terrestrial Integrated Networks". In: *IEEE/ACM Trans. Netw.* 32.5 (2024), pp. 3840–3853.

[127] Y. Li, L. Liu, H. Li, W. Liu, Y. Chen, W. Zhao, J. Wu, Q. Wu, J. Liu, and Z. Lai. "Stable Hierarchical Routing for Operational LEO Networks". In: *Proc. ACM MobiCom*. 2024, pp. 296–311.

[128] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, and X. Chen. "Shortest Path in LEO Satellite Constellation Networks: An Explicit Analytic Approach". In: *IEEE J. Sel. Areas Commun.* 42.5 (2024), pp. 1175–1187.

[129] Y. Huang, B. Feng, A. Tian, P. Dong, S. Yu, and H. Zhang. "An Efficient Differentiated Routing Scheme for MEO/LEO-Based Multi-Layer Satellite Networks". In: *IEEE Trans. Netw. Sci. Eng.* 11.1 (2024), pp. 1026–1041.

[130] P. Zuo, C. Wang, Z. Yao, S. Hou, and H. Jiang. "An Intelligent Routing Algorithm for LEO Satellites Based on Deep Reinforcement Learning". In: *Proc. IEEE VTC*. 2021, pp. 1–5.

[131] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, S. Sun, X. Shen, and H. V. Poor. "Interactive AI With Retrieval-Augmented Generation for Next Generation Networking". In: *IEEE Netw.* 38.6 (2024), pp. 414–424.

[132] S. Li, Q. Wu, and R. Wang. "Efficient Packet Routing in Ultra-Dense LEO Satellite Networks via Cooperative-MARL with Queuing Theory Model". In: *Proc. IEEE WCNC*. 2025, pp. 1–6.

[133] R. Zhang, K. Xiong, Y. Lu, P. Fan, D. W. K. Ng, and K. B. Letaief. "Energy Efficiency Maximization in RIS-Assisted SWIPT Networks With RSMA: A PPO-Based Approach". In: *IEEE J. Sel. Areas Commun.* 41.5 (2023), pp. 1413–1430.

[134] K. He, T. X. Vu, D. T. Hoang, D. N. Nguyen, S. Chatzinotas, and B. E. Ottersten. "Risk-Aware Antenna Selection for Multiuser Massive MIMO Under Incomplete CSI". In: *IEEE Trans. Wirel. Commun.* 23.9 (2024), pp. 11001–11014.

[135] J. Song, Y. Ju, L. Liu, Q. Pei, C. Wu, M. A. Jan, and S. Mumtaz. "Trustworthy and Load-Balancing Routing Scheme for Satellite Services with Multi-Agent DRL". In: *Proc. IEEE INFOCOM Workshop*. 2023, pp. 1–6.

[136] Q. Yang, T. D. Simão, S. H. Tindemans, and M. T. Spaan. "WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning". In: *Proc. AAAI*. Vol. 35. 12. 2021, pp. 10639–10646.

[137] K. Menda, Y. Chen, J. Grana, J. W. Bono, B. D. Tracey, M. J. Kochenderfer, and D. H. Wolpert. "Deep Reinforcement Learning for Event-Driven Multi-Agent Decision Processes". In: *IEEE Trans. Intell. Transp. Syst.* 20.4 (2019), pp. 1259–1268.

[138] J. Lee, J. Park, M. Bennis, and Y. Ko. "Integrating LEO Satellites and Multi-UAV Reinforcement Learning for Hybrid FSO/RF Non-Terrestrial Networks". In: *IEEE Trans. Veh. Technol.* 72.3 (2023), pp. 3647–3662.

[139] A. Chaaban, Z. Rezki, and M. Alouini. "On the Capacity of Intensity-Modulation Direct-Detection Gaussian Optical Wireless Communication Channels: A Tutorial". In: *IEEE Commun. Surv. Tutorials* 24.1 (2022), pp. 455–491.

[140] M. Fonoberova and D. Lozovanu. "Optimal multicommodity flows in dynamic networks and algorithms for their finding". In: *Buletinul Academiei de Ştiinţe a Republicii Moldova. Matematica* 47.1 (2005), pp. 19–34.

[141] W. Jung, M. Cho, J. Park, and Y. Sung. "Quantile Constrained Reinforcement Learning: A Reinforcement Learning Framework Constraining Outage Probability". In: *Adv. Neural Inf. Process. Syst.* 2022.

[142] Q. Yang, T. D. Simão, S. H. Tindemans, and M. T. J. Spaan. "Safety-constrained reinforcement learning with a distributional safety critic". In: *Mach. Learn.* 112.3 (2023), pp. 859–887.

[143] W. Dabney, G. Ostrovski, D. Silver, and R. Munos. "Implicit Quantile Networks for Distributional Reinforcement Learning". In: *Proc. ICML*. Vol. 80. 2018, pp. 1104–1113.

[144] Z. Chen, Y. Zhou, and H. Huang. "On the duality gap of constrained cooperative multi-agent reinforcement learning". In: *Proc. ICLR*. 2024.