

Multi-Encoder Semantic Communication for Human Digital Twin Synchronization

Oluwasegun Talabi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

December 2024

© Oluwasegun Talabi, 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Oluwasegun Talabi**

Entitled: **Multi-Encoder Semantic Communication for Human Digital Twin Synchronization**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Dongyu Qiu

_____ External Examiner
Dr. Wen-Fang Xie

_____ Supervisor
Dr. Jun Cai

Approved by _____
Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

_____ 2024

Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Multi-Encoder Semantic Communication for Human Digital Twin Synchronization

Oluwasegun Talabi

Human Digital Twin (HDT) concept introduces an innovative framework for creating digital counterparts of individuals, enabling real-time synchronization between the physical twin (PT) and the virtual twin (VT). This PT-VT synchronization underpins various human-centered services but demands significant data processing and efficient communication resource allocation, particularly in resource-constrained environments. Semantic communication has emerged as a promising alternative to traditional data-driven methods; however, single-encoder models struggle to meet the diverse and dynamic requirements of HDT applications.

To address these challenges, this thesis presents a multi-encoder semantic communication framework that adaptively allocates resources—such as bandwidth, computational power, and processing frequency—based on application-specific needs. The short-term optimization problem is formulated as a mixed-integer nonlinear programming (MINLP) problem and solved using a genetic algorithm (GA). Simulation results demonstrate that the proposed multi-encoder model significantly improves synchronization quality and power efficiency, outperforming traditional single-encoder models in terms of accuracy, latency, and resource utilization.

To achieve a balance between immediate performance and long-term objectives (e.g., queue stability, sustainable energy usage, and high throughput), the thesis explores the long-term optimization problem, formulated as a Markov Decision Process (MDP) and solved using Lyapunov-assisted deep reinforcement learning. This adaptable and scalable approach positions the multi-encoder semantic communication model as a highly efficient solution for future HDT applications.

Acknowledgments

First and foremost, I express my heartfelt gratitude to God Almighty for His abundant grace, guidance, and blessings that have enabled me to complete this thesis. Without His strength, this journey would not have been possible.

I am deeply thankful to my supervisor, Dr. Jun Cai, for his invaluable support, insightful advice, and encouragement throughout the course of this work. His expertise and guidance have been instrumental in shaping this thesis.

To my beloved wife Nkechi, your unwavering love, patience, and encouragement have been my greatest motivation. Thank you for your sacrifices and for standing by me every step of the way.

To my parents and family, your prayers, support, and belief in me have been my foundation. I am forever grateful for your love and sacrifices.

Finally, to my friends and colleagues, thank you for your encouragement, collaboration, and companionship during this journey. Your support has meant more than words can express.

This work is a reflection of the love and support I have received from all of you, and I am truly blessed to have you in my life.

Contents

List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Background	1
1.2 Objectives	5
1.3 Contributions	6
1.4 Thesis Structure	7
2 Literature Review	8
2.1 Digital Twin to Human Digital Twin	8
2.2 Traditional to Semantic Communication in Digital Twin Framework	15
3 Multi-Encoder Semantic Communication for Human Digital Twin Synchronization	21
3.1 System Model	21
3.1.1 Semantic Encoder	23
3.1.2 Transmission Model	24
3.2 Problem Formulation	25
3.3 Genetic Algorithm Based Solution	28
3.4 Simulation Results	31

4 Lyapunov-Assisted DRL-based Multi-Encoder Semantic Communication for HDT Synchronization	37
4.1 System Model	38
4.1.1 Queuing Model	38
4.2 Problem Formulation	39
4.2.1 Lyapunov Guided Problem Transformation	43
4.3 Deep Reinforcement Learning based Online HDT Synchronization	46
4.3.1 Model Free Actor Module	47
4.3.2 Model-based Critic Module	48
4.3.3 DNN Policy Update Module	52
5 Conclusion and Future Work	54
5.1 Conclusion	54
5.2 Future Work	55
5.2.1 Developing and Simulating the framework "Lyapunov-Assisted DRL-based Multi-Encoder Semantic Communication for HDT Synchronization" as outlined in Chapter 4	55
5.2.2 Preemptive Queue Protocol	55
Bibliography	57

List of Figures

Figure 2.1	Simple Digital Twin Model	9
Figure 3.1	System Model	22
Figure 3.2	The Latency Violation versus the number of inner iterations n	33
Figure 3.3	The Accuracy Violation versus the number of inner iterations n	34
Figure 3.4	The Power Consumption versus the number of inner iterations n	35
Figure 3.5	The Average Total Reward R versus the number of inner iterations n	36
Figure 4.1	DRL framework schematic	46

List of Tables

Table 3.1 Simulation Parameters 32

Chapter 1

Introduction

This chapter provides essential background information to understand the concept and importance of multi-encoder semantic communication in Human Digital Twin synchronization.

1.1 Background

The concept of a Human Digital Twin (HDT) represents a transformative step in digital human representation, extending the capabilities of digitalization to offer highly personalized, human-centric services across diverse fields, such as healthcare, urban infrastructure, transportation, and next-generation 6G networks. This pioneering approach opens up avenues for customized healthcare, efficient city management, and adaptive network structures centered around human needs. Essentially, the HDT is a digital replica of an individual, created through extensive data collection from sources like pervasive sensor networks, electronic health records, and other interconnected systems [1]. This data is transmitted and integrated into a digital environment, where the HDT can mirror an individual's state and continuously adapt to real-world changes. Analogous to traditional digital twin (DT) technology, which has been widely applied in industries to simulate and monitor physical assets, HDT enables continuous data exchange that reflects real-time shifts in a person's health and environment. This digital replica provides proactive insights, enabling predictive assessments of health based on the latest data [2]. The HDT's role is particularly impactful in personalized healthcare services (PHS), where it leverages artificial intelligence (AI) to facilitate patient-centered

care. Through sophisticated algorithms, AI processes the diverse data streams collected by the HDT to deliver customized, data-informed guidance that aids both individuals and healthcare providers in making informed decisions [2]. By integrating information from individual medical histories, genetic data, lifestyle factors, and personal preferences, HDT technology supports preventive care initiatives and the design of treatment regimens tailored to the unique characteristics of each patient [3]. This personalized approach not only enhances patient outcomes but also empowers healthcare practitioners by providing them with rich, multidimensional insights that improve the quality of life for individuals, fostering a more precise and responsive healthcare model.

An HDT system is structured around three fundamental components that collaboratively bridge the physical and digital realms: the Physical Twin (PT), which represents the actual human in the real world; the Virtual Twin (VT), which is the digital reflection of the PT; and the advanced data links that enable continuous information exchange between these two entities. This tripartite setup is designed to provide a holistic representation of individuals in a digital format, paving the way for various applications in personalized healthcare, smart city management, and adaptive network services. By establishing a seamless interaction between PT and VT, the HDT system can capture and respond to real-world changes in near-real-time, ensuring that the digital twin mirrors the individual's current state as closely as possible [4]. The synchronization of PT and VT must be both accurate and timely to achieve optimal outcomes, particularly in healthcare where real-time updates and precise monitoring can significantly impact patient care. Since the synchronization process is data-intensive and highly sensitive to delays, an effective communication framework becomes essential. This framework is responsible for managing the constant data flow required to align the PT and VT while also ensuring efficient resource utilization. The delay-sensitive nature of synchronization demands a system that minimizes latency and maximizes the reliability of data transmission. With the growth of HDT applications in healthcare, optimizing communication channels to handle large datasets quickly and effectively is a crucial priority [1] [5].

In recent years, semantic communication has emerged as an innovative solution to the inherent challenges of traditional synchronization methods. Unlike conventional approaches that rely on the transmission of vast amounts of raw data, semantic communication selectively transmits only the information deemed essential for maintaining PT and VT alignment. By transmitting meaning rather

than raw data, semantic communication can significantly reduce the volume of data exchanged between the PT and VT. This results in lower bandwidth consumption, reduced latency, and enhanced synchronization efficiency, allowing the HDT system to operate more effectively while using fewer resources [6].

In reality, the frequency and precision of PT-VT synchronization vary based on HDT application needs [7]. For example, real-time monitoring of critical metrics like heart rate demands high-frequency, low-latency updates, whereas sleep tracking can accommodate more relaxed constraints. This diversity underscores that a one-size-fits-all communication solution would be impractical and inefficient for HDT systems. Implementing a uniform semantic communication system across various HDT applications may lead to resource wastage in simpler applications (such as sleep monitoring) that don't require high-frequency updates. On the other hand, a generic system might not be able to meet the stringent demands of more critical, real-time applications that rely on rapid data exchange and high accuracy. A flexible and efficient semantic communication system that can dynamically adapt to the unique requirements of each HDT application is therefore essential. Such a system would adjust its communication frequency, data precision, and processing power based on the specific application's needs. For instance, it could prioritize low-latency, high-fidelity data transfers for critical monitoring while conserving resources by reducing update frequency and precision for non-urgent applications. This adaptive approach optimizes resource utilization, minimizes unnecessary energy consumption, and ensures that each HDT application receives the appropriate level of synchronization, leading to both improved system efficiency and a better overall user experience.

Furthermore, semantic communication relies extensively on machine learning algorithms to interpret and filter relevant information before transmission, a process that is computationally and energy-intensive [8]. Machine learning models used in semantic communication, particularly deep learning models, often require substantial processing power to analyze data in real-time and extract meaningful insights. This computational demand poses a challenge in HDT systems, where many components, such as on-body IoT sensors, are constrained by limited processing capabilities and battery life. For instance, wearable devices and other on-body sensors used in HDT applications are typically designed for low-power operation, prioritizing small size and lightweight form factors. These devices often lack the necessary computational resources to handle complex semantic

processing on-site. Implementing a dedicated semantic transmitter for each PT component would therefore be impractical, as the processing power required would quickly drain battery resources, increase device size, and ultimately compromise the usability and lifespan of these devices. This limitation suggests that HDT systems must consider alternative approaches to enable semantic communication without overburdening individual components.

The challenges posed by diverse HDT application requirements highlight the need for a multi-encoder semantic communication system. Unlike a single-encoder setup, a multi-encoder system is designed to manage multiple streams of data, each tailored to specific applications, thereby enhancing the flexibility and responsiveness of data transmission. However, this shift to a multi-encoder architecture significantly increases the complexity of resource management. Such a system must handle real-time, adaptive distribution of critical resources—such as bandwidth, computational power, and energy—across encoders that are configured differently based on each application’s demands. In practice, HDT applications vary widely in their requirements, and meeting these demands requires sophisticated resource management strategies. These strategies must account for the diverse data rates, latency requirements, and processing complexities of each encoder, adjusting resource allocation dynamically to avoid either resource wastage or bottlenecks. Moreover, in environments where multiple HDT transmitters are actively sharing network resources, the system needs to prioritize and balance demands to prevent network congestion. This is especially important for ensuring reliable, real-time data aggregation and transmission across applications, even when operating under the high loads of concurrent data streams.

Overall, the multi-encoder approach not only accommodates the heterogeneous nature of HDT applications but also pushes the boundaries of current resource management frameworks, requiring advanced strategies that can adaptively respond to fluctuating demands in real-time.

Previous studies have investigated semantic communication and resource allocation within communication networks and DT systems, typically focusing on single-encoder models [6] [9] [10]. These approaches, however, fall short in addressing the dynamic resource needs of HDT systems, where each PT update may have unique synchronization demands, creating a need for more adaptable and responsive resource management.

1.2 Objectives

The objective of this thesis work is to develop a system model for a multi-encoder semantic communication framework designed to meet the dynamic requirements of HDT update data. This involves several key steps:

- **System Modeling:** Create a detailed system model of a multi-encoder semantic communication setup specifically tailored to handle the unique and variable synchronization needs of HDT data updates. This model will capture the behavior of multiple encoders working in tandem to process and transmit HDT data based on real-time demands.
- **Mathematical Formulation:** Abstract the developed system model into a coherent mathematical framework, formulating the problem in terms of resource allocation, communication efficiency, and synchronization requirements. This mathematical formulation will capture the interactions and constraints of the multi-encoder system under resource limitations.
- **Optimization Equation Formulation:** Derive an optimization equation that addresses resource constraints within the HDT system. This equation will target optimal allocation of bandwidth, computational power, and energy resources to maximize performance metrics such as data accuracy, latency, and responsiveness.
- **Solution Development:** Examine the characteristics of the formulated equation and its constraints to identify a suitable solution approach. Then, solve the formulated optimization problem using appropriate optimization methods.
- **Simulation and Result Interpretation:** Conduct simulations to assess the system's performance across various conditions, focusing on how effectively it meets HDT update requirements while optimizing resource usage. Analyze the simulation results, emphasizing the system's efficiency and responsiveness. Compare these results with existing single-encoder models and other state-of-the-art approaches to highlight the advantages and potential improvements offered by the multi-encoder framework.

By following these steps, this thesis aims to contribute a novel, adaptable solution to resource management in HDT systems, addressing the gaps in existing single-encoder models and providing

insights into the performance benefits of a multi-encoder approach.

1.3 Contributions

To meet the application requirement of HDT, this thesis work proposes a novel multi-encoder semantic communication framework designed for HDT synchronization. This framework leverages multiple dynamic encoders and shared network resources to optimize utilization, adjusting synchronization parameters such as update frequency and accuracy based on each application's specific requirements. By employing semantic encoders with varied configurations in a multi-encoder setup, HDT update data are dynamically allocated according to specific needs and encoder capabilities, with resources then assigned to meet predefined HDT requirements. This strategy ensures an efficient balance between synchronization quality and resource costs, enabling even resource-constrained devices to integrate effectively within HDT networks and supporting the scalable, reliable deployment of personalized healthcare solutions. To my knowledge, this is the first framework to meet the intricate requirements of HDT semantic communication with a multi-dynamic encoder approach. This pioneering model sets a new standard in resource management, providing a scalable and adaptable solution for future applications. The main contributions of this paper are summarized as follows:

- We propose multi-encoder semantic encoder for HDT that leverages shared network resources for encoding and transmission, thereby optimizing the connection between PT and VT to meet individual application needs.
- We analyze the incident arrival pattern to efficiently allocate HDT update data to encoders that meet its application requirements and maximize the system reward. Using a multi-dynamic encoder setup and shared network resources, we quantify accuracy, latency, and overall operational power cost for each user allocation. System performance is then evaluated through a linear combination of latency and achieved accuracy.
- We formulate a short-term performance maximization problem as a mixed-integer nonlinear

programming (MINLP) problem and we solve the formulated problem using genetic algorithm (GA). This approach effectively meets the dynamic needs of individual HDT applications through efficient resource allocation. Simulation results demonstrate the superiority of the proposed model compared to existing frameworks.

- We proceed to formulate a long-term stochastic MINLP problem with a long-term queue and power constraint. We first reduce the problem to a single-term problem using the Lyapunov approach and then solve the single-term markov decision problem (MDP) using a model-free actor and model-based critic deep reinforcement model.

It is worth noting that the short-term problem of this work has been compiled into a research paper and submitted for review and publication in IEEE International Conference on Communications scheduled for 8–12 June 2025 here in Montreal. Simulation is ongoing for the long-term problem, it will also be compiled and submitted for review and publication in IEEE journal.

1.4 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 provides a review of related works, offering an overview of existing research in the field. Chapter 3 introduces the proposed system model, detailing its framework and components, followed by an analysis of system performance and the formulation of the corresponding performance optimization problem. This problem is solved using a genetic algorithm, and the solution is then simulated and compared with existing models. Chapter 4 extends the short-term optimization problem from Chapter 3 to a long-term optimization framework. This extended problem is addressed using Lyapunov-assisted deep reinforcement learning. Finally, chapter 5, summarizes the research findings and discusses the implications of the work.

Chapter 2

Literature Review

2.1 Digital Twin to Human Digital Twin

The concept of the digital twin (DT), while not entirely new, has recently surged in popularity. As described in Figure 2.1 it's simply a virtual model that replicates a physical entity through real-time data exchange to enable monitoring, simulation, and optimization. Its early roots can be traced to 1991 when David Gelernter introduced the idea of "Mirror Worlds," a software model intended to reflect the physical world through real-world data. In his book [11], Gelernter outlines five essential components for developing a mirror world which includes: a "deep picture," a "live picture," agents, history, and the integration of these elements into a cohesive system. In the context of Product Life-cycle Management (PLM), Michael Grieves introduced the concept of the "Mirrored Spaces Model" at the University of Michigan in 2002. This model included three primary components: real space, virtual space, and a linking mechanism that facilitated data and information flow between them[12]. In 2006, the model was renamed to the "Information Mirroring Model". The term DT first appeared in NASA's draft technological roadmap in 2010, where it was also called "Virtual Digital Fleet Leader" [13]. The concept was described as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its physical counterpart" Since NASA's initial definition, several authors have described and explored the concept and potential of Digital Twin (DT), with applications spanning manufacturing[14][15][16], transportation[17][18], urban planning[19][20], and

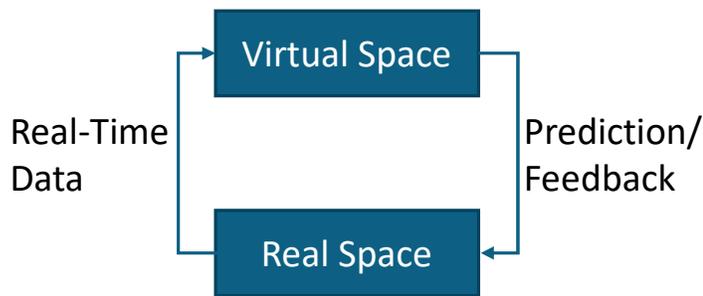


Figure 2.1: Simple Digital Twin Model.

many other sectors.

Recently, the scope of Digital Twin (DT) technology has expanded beyond machines, systems, and industrial applications to include human beings, this concept is termed Human Digital Twin (HDT). This extension into human-focused DTs leverages real-time data and detailed modeling to simulate and analyze individual health, behavior, and responses to various conditions. In health-care, for instance, a human Digital Twin can be used to monitor physiological parameters, predict potential health risks, and create personalized treatment plans based on real-time data[3]. HDT has garnered significant attention from different directions in recent years, domains including sport, health, and manufacturing have all benefited from this research effort[21]. like other VT, HDT is essentially a virtual model that mirrors the physical and sometimes psychological states of a human being[1]. This dynamic digital replica is updated continuously with real-time data, enabling comprehensive monitoring, predictive analytics, and personalized interventions. The resourcefulness of HDTs lies in their ability to address both current and potential challenges effectively, by leveraging continuous data collection from various sources, including wearable devices, medical records, genetic profiles, and environmental sensors [1], HDTs provide a holistic view of an individual’s status. This comprehensive monitoring capability allows for early detection of anomalies, timely interventions, and personalized plans tailored to the specific needs of each individual. The anticipatory nature of HDTs is particularly valuable in predicting future problems and preventing them before they become critical. In HDT, each human VT continuously evolves by integrating data from its physical counterpart (PT), which resides in the real world. This dynamic and data-driven evolution introduces significant challenges in the design and implementation of human VTs. These challenges

stem from the need to accurately capture the intricate and variable nature of human behaviors, interactions, and responses within their physical context, making the process far more complex than that of conventional DTs.

In [22], Barricelli et al. present an innovative approach to fitness management through the implementation of HDT. This study emphasizes the potential of HDT in transforming personalized fitness and health monitoring by utilizing advanced technologies such as artificial intelligence, Internet of Things (IoT), and big data analytics to capture and analyze real-time data from wearable sensors and mobile applications. The authors explained that the HDT framework facilitates a continuous and dynamic representation of an individual's health, allowing for the precise modeling of physiological states and behaviors. This capability enhances fitness management by enabling personalized and adaptable health interventions based on an individual's unique data profile. In line with this, Barricelli et al. discuss the technical architecture required to support HDTs, including the integration of data sources and the importance of real-time processing. Their findings suggest that HDTs can predict and recommend tailored fitness plans, providing users with insights to improve health outcomes and optimize personal fitness goals.

In addition to technical considerations, Barricelli et al. explore the ethical and privacy challenges associated with HDT deployment. Given the sensitivity of health-related data, the authors highlight the critical need for robust data governance frameworks to protect user privacy and ensure data security, which they identify as essential for fostering user trust and regulatory compliance. This study thereby contributes to the growing body of literature on digital health innovations by emphasizing the dual importance of technical innovation and ethical standards in deploying effective, user-centered digital twins for fitness and health applications.

Wang et al. in [23] investigate the application of DT technology in human-robot interactive welding systems, with a focus on analyzing welder behavior. This study introduces a DT model designed to synchronize virtual and physical entities in real-time, creating an interactive environment where both human and robot actions are mirrored and monitored within a digital replica. The DT model leverages sensors and data analytics to simulate welding processes and track both robotic and welder performance, providing insights into behavior and skill assessment. The authors highlight the role of DT in enhancing collaboration between human welders and robots by capturing detailed

process data, such as welder posture, movement, and technique, which is used to optimize welding quality and productivity. Real-time data integration enables the DT system to detect anomalies and improve decision-making by providing feedback for both automated and manual adjustments. Additionally, the study explores how welder behavior analysis through DT can support training by identifying areas for skill improvement.

Wang et al. also address challenges related to implementing DT, such as data processing demands and the need for high synchronization accuracy between physical and digital models. This research contributes to the growing field of human-robot interaction by demonstrating the DT's potential in optimizing complex, skill-intensive industrial tasks like welding, thereby advancing both productivity and worker safety.

In [24] Graessler and Pöhler investigate the role of DT in representing human operators within the scheduling processes of a cyber-physical production system (CPPS). Their study explores how integrating a human-centered DT into CPPS scheduling can optimize production workflows by accounting for human factors such as availability, skills, and work patterns. The authors propose a model in which the DT serves as a virtual representation of human operators, allowing for real-time adjustments in production scheduling based on operator status and capabilities. This integration supports more flexible, adaptive scheduling by enabling the system to dynamically reassign tasks based on human operator conditions. The model aims to enhance production efficiency by ensuring that both machines and humans are utilized optimally within the manufacturing environment. Through this integration, Graessler and Pöhler demonstrate the potential of DTs to bridge human and machine workflows in CPPS, highlighting how HDT can be applied to improve scheduling accuracy, reduce downtime, and enhance the adaptability of production systems in response to real-time operational conditions.

Liu et al in [25] present a novel cloud-based framework for elderly healthcare services utilizing DT technology, aimed at enhancing healthcare delivery for aging populations. Their study introduces a system where elderly individuals are represented by digital replicas that monitor and manage health metrics in real-time. These DTs are connected to a cloud platform, enabling continuous data collection and analysis from wearable devices, health sensors, and other IoT-enabled tools. The proposed framework allows for proactive healthcare management by leveraging the DT

to track vital signs, physical activities, and other health-related parameters. The system is designed to provide personalized health interventions, alerting healthcare providers or family members in case of abnormal readings or emergencies. By integrating cloud computing, the framework ensures scalability and accessibility, allowing for real-time health monitoring and decision-making support regardless of geographical location.

Liu et al. emphasize the potential of this cloud-based DT approach to improve the quality of elderly care, enhancing independence, safety, and overall well-being. The study highlights the importance of this technology in addressing the healthcare challenges posed by an aging population, offering a flexible, efficient solution that integrates advanced monitoring and real-time data analytics for effective health management.

In [26] Laaki, Miche, and Tammi explore the development of a DT prototype designed for real-time remote control applications over mobile networks, focusing specifically on remote surgery. Their study addresses the technical requirements for a DT system that can facilitate real-time, high-precision control necessary for surgical procedures conducted at a distance. The prototype leverages mobile network infrastructure to support low-latency, reliable data transmission between the DT and physical systems. The authors detailed how this DT model can simulate and control remote surgical tools, providing real-time feedback to the operator and allowing for precise adjustment. The application of DT technology in this context enables surgeons to perform delicate procedures remotely by offering a virtual replica that mirrors the actions of surgical instruments in real-time. Key challenges discussed include the need for ultra-low latency and high network reliability to ensure safety and accuracy in remote operations.

The paper demonstrates that the DT prototype has the potential to expand the reach of specialized healthcare services, enabling skilled professionals to operate on patients in distant or underserved locations. This study contributes to the field by underscoring the role of DTs in enhancing remote healthcare applications, particularly in areas where real-time responsiveness and precision are critical.

Zhang et al. in [27] examine the application of DT technology in healthcare, focusing on enhancing cyber resilience for lung cancer care. The study proposes a DT framework that digitally mirrors a patient's lung cancer progression and treatment response, aiming to improve both patient

care and system security. The DT integrates real-time patient data, such as imaging and diagnostic information, creating a dynamic model to support personalized treatment planning and continuous monitoring. The paper emphasizes cyber resilience within this DT framework to address the growing concerns of data security and reliability in healthcare applications. The authors discussed strategies to ensure data integrity and protect sensitive health information, critical for maintaining trust and compliance in healthcare systems. By employing robust cybersecurity measures, the DT model seeks to prevent disruptions or data breaches that could compromise patient outcomes or hinder healthcare delivery.

This study contributes to the intersection of digital health and cybersecurity, highlighting the importance of cyber resilience for safe and effective DT deployment in healthcare. Zhang et al. showcase how a cyber-resilient Digital Twin can provide precise, adaptive support for lung cancer patients while safeguarding against potential cyber threats, ultimately improving both clinical care and system robustness.

In [28] Hu et al. provide a comprehensive review of Driver Digital Twin (DDT) technology and its potential to advance intelligent vehicle systems. The Driver Digital Twin is a real-time digital replica of a driver, designed to monitor, analyze, and predict driver behaviors to improve vehicle safety and performance. The authors explore how DDT can enhance driver-vehicle interactions by providing insights into driver states, such as alertness, attention, and behavior patterns, which are critical for accident prevention and adaptive vehicle responses. The paper discusses enabling technologies for DDT, including sensors, machine learning algorithms, and communication systems, that facilitate the real-time data collection and analysis needed to mirror driver actions and conditions. By leveraging this data, intelligent vehicles equipped with DDT can anticipate driver actions, adjust to behavioral changes, and implement safety measures proactively.

Hu et al. also highlight the potential applications of DDT in autonomous driving, where it could assist in transitioning control between human drivers and autonomous systems smoothly and safely. The authors conclude by discussing challenges such as data privacy, real-time data processing, and cybersecurity, emphasizing the importance of robust frameworks to ensure DDT's safe deployment in intelligent vehicles. This review underscores the DDT's role in enhancing intelligent transportation systems, bridging human drivers and autonomous technologies for safer and more adaptive

vehicles.

Martinez-Velazquez, Gamez, and El Saddik in [29] introduce the "Cardio Twin," a DT of the human heart, developed to operate on edge computing infrastructure for real-time cardiovascular monitoring. This study aims to enhance cardiac health monitoring by creating a responsive, high-fidelity digital replica of a patient's heart that functions at the network edge, closer to data sources like wearable devices and medical sensors. The Cardio Twin system leverages real-time physiological data to model the heart's condition and functions dynamically, enabling early detection of abnormalities and immediate feedback on heart health. Running on edge devices minimizes latency, making the DT capable of real-time response, critical for applications such as remote patient monitoring and emergency interventions.

The authors discuss the advantages of edge computing for DTs in healthcare, emphasizing how processing data near the source improves both speed and data security. By reducing dependence on cloud infrastructure, the Cardio Twin can operate effectively even with limited internet connectivity, making it particularly useful in remote or underserved locations. Martinez-Velazquez et al. showcase the potential of edge-based Digital Twins like the Cardio Twin to advance personalized, timely healthcare, especially for managing chronic conditions such as cardiovascular disease.

In [30] Elayan, Aloqaily, and Guizani proposed a DT framework designed to enhance intelligent, context-aware IoT healthcare systems. The study focuses on using DT technology to create dynamic, real-time digital replicas of patients within IoT-based healthcare environments, enabling continuous monitoring and adaptive responses to individual health needs. The DT framework integrates data from IoT-enabled sensors and devices to build a personalized model that reflects a patient's current health status and environmental context. This system allows for context-aware decision-making, where healthcare interventions are tailored based on both the patient's physiological data and their surroundings, such as activity levels, location, and environmental factors. Considering heart condition diagnostics through a highly accurate ECG classifier model integrated with Neural Network(NN), the authors highlight the role of AI and machine learning within the DT framework to process and analyze the vast data generated by IoT devices, facilitating real-time adjustments in healthcare management. They also discuss challenges related to data privacy, interoperability, and security in IoT healthcare systems. Elayan et al. conclude that by incorporating

context awareness and adaptability, DTs can significantly improve the responsiveness and personalization of IoT-based healthcare, supporting proactive and efficient health management in real-time.

Amara, Kerdjadj, and Ramzan in [31] explore the use of emotion recognition to develop an affective HDT using virtual reality (VR) technologies. This study aims to enhance the HDT model by incorporating emotional states, making it possible to create a digital replica that mirrors not only physical but also emotional attributes in real-time. The VR-enabled HDT can capture and process human emotional responses through techniques like facial expression analysis, physiological signals, and contextual cues within immersive VR environments. The authors propose that by integrating emotion recognition, HDT can better simulate and respond to human affective states, which has applications in areas like personalized mental health support, human-computer interaction, and adaptive training programs. This approach allows the HDT to provide customized feedback, adapt interactions according to the user's mood, and potentially improve user engagement by responding empathetically to emotional cues. The paper also discusses the technological requirements for real-time emotion recognition, such as advanced sensors and machine learning algorithms, which enable HDT to continuously assess and update the user's emotional state. The study contributes to the development of affective computing and highlights the potential of emotion-sensitive HDTs to create more responsive, intuitive, and human-like digital twins that enhance the user experience in virtual and augmented environments.

2.2 Traditional to Semantic Communication in Digital Twin Framework

Traditional communication in digital twins generally involves the transmission of large amounts of raw data between the physical and virtual environments. This data includes real-time sensor readings, status updates, control commands, and historical data, which is then processed and analyzed to synchronize and simulate the digital twin. Numerous research works have been dedicated to the optimization of this model of communication to improve transmission efficiency and minimize latency. These optimizations aim to enhance the responsiveness and scalability of DT systems, especially in applications requiring real-time decision-making and interaction. In [32] Okegbile and Cai

propose an edge-assisted connectivity scheme to enhance HDT frameworks by ensuring efficient human-to-virtual twin connections and using blockchain and federated learning to enhance security and privacy. The study focuses on utilizing edge computing to support low-latency, high-reliability communication between individuals and their digital twins, enabling real-time data exchange and interaction. The proposed scheme leverages edge devices to process data locally, reducing reliance on cloud infrastructure and minimizing delays that could hinder the HDT's responsiveness. This edge-assisted approach is particularly valuable for applications requiring low latency and high levels of interactivity, such as remote health monitoring, augmented reality, and real-time decision-making. The HDT framework can provide faster, context-aware responses, enhancing user experience and reliability by maintaining connectivity at the network edge. The paper highlights the technical challenges associated with edge-assisted HDTs, such as resource management, data security, and interoperability. Their scheme addresses these issues by optimizing data handling protocols and ensuring seamless integration across diverse network environments. This research underscores the potential of edge computing in advancing HDT frameworks, emphasizing its role in achieving responsive, scalable, and secure connections for real-time digital twin applications.

Okegbile et al. in [4] presented a differentially private federated multi-task learning framework designed to enhance human-to-virtual connectivity within HDT systems. This framework addresses the dual goals of data privacy and efficient connectivity by using federated learning to enable decentralized data processing, which ensures sensitive personal data remains secure on local devices. The proposed framework supports multi-task learning, allowing HDTs to handle diverse tasks, such as health monitoring, behavioral analysis, and personalized recommendations, without compromising user privacy. By implementing differential privacy, the framework protects individual data contributions from being identifiable, which is crucial in healthcare and other applications where confidentiality is paramount. This privacy-preserving setup enhances trust and compliance in HDT systems, encouraging broader adoption of DT technologies in sensitive domains. The paper demonstrates that this federated learning approach, combined with differential privacy, provides a scalable and secure solution for continuous, real-time human-to-virtual twin connectivity. The study highlights the importance of privacy-preserving techniques in advancing HDTs, especially in applications where personal data security and multi-functionality are essential for effective digital

twin operations.

The authors have addressed the challenge of allocating communication resources in HDT systems; however, they have yet to address the critical issue of information bottlenecks resulting from the high communication overhead due to large data volumes. These bottlenecks can significantly hinder data transmission and processing efficiency, impacting both the accuracy and latency of HDT systems. To mitigate these challenges, Semantic Communication (SC) has been integrated into DT and HDT systems. SC not only reduces communication overhead in HDT networks but also improves contextual understanding and accuracy through its machine-learning framework. In [33] Li et al. introduce a semantic-enhanced DT system designed to monitor interactions between robots and their environments. This system leverages semantic technologies to create a DT that replicates a robot's physical state and environment and interprets the meaning of interactions. By adding semantic layers, the DT can provide contextual insights into how a robot perceives and responds to its surroundings. The semantic-enhanced DT system integrates sensor data with semantic models, enabling it to analyze and understand complex interactions in real-time. This capability improves monitoring accuracy and responsiveness in applications where robots must adapt to dynamic environments, such as autonomous navigation, manufacturing, and service robotics. The system allows for more intelligent decision-making by categorizing and interpreting interaction patterns, which in turn supports predictive maintenance, safety monitoring, and adaptive responses. The paper emphasizes the value of adding semantic understanding to DTs, as it enables robots to interact with their environments in more meaningful ways, enhancing both functionality and adaptability. Their study highlights the potential of semantic-enhanced DTs to advance robot-environment interaction monitoring, particularly in settings where situational awareness and context-sensitive responses are critical.

In [34] Du et al. propose a YOLO-based SC framework enhanced with generative AI-aided resource allocation to support the construction of DTs. This study introduces a system that leverages YOLO object detection for semantic understanding, combined with generative AI to optimize resource allocation in DT construction, improving both efficiency and precision in real-time data processing. The YOLO-based approach enables the system to recognize and prioritize important objects and events within raw data, facilitating focused communication that emphasizes meaningful

content. Generative AI further enhances this framework by dynamically allocating resources according to DTs' needs, ensuring sufficient computational power and bandwidth are directed where most impactful. This integration allows for efficient DT updates, reducing unnecessary data transmission and enabling responsive, high-fidelity DT models.

The authors highlight the advantages of combining YOLO's rapid object detection with generative AI's adaptive resource management, particularly in complex IoT environments requiring real-time interaction and analysis. The framework's ability to identify semantic relevance and allocate resources efficiently supports applications like smart cities, industrial monitoring, and autonomous systems, where precise and responsive DTs are essential. This study demonstrates the potential of AI-enhanced semantic communication to streamline DT construction, promoting effective data utilization and resource efficiency across diverse IoT contexts.

Thomas et al. in [6] explore a causal semantic communication framework for DTs, applying a generalizable imitation learning approach. This study introduces a method for enhancing the communication between DTs and their physical counterparts by incorporating causal semantic reasoning, allowing the DTs to interpret and prioritize information based on causality and relevance rather than just raw data transmission. The authors employ imitation learning to train DTs in recognizing causal relationships within data streams, enabling more efficient and context-aware information sharing. This approach reduces communication overhead by focusing on transmitting only the most relevant, causally significant information, which is essential in applications where bandwidth and processing resources are limited. The DTs are thus able to communicate in a more human-like, semantic manner, improving their ability to interpret complex systems and make informed decisions in real-time. The authors demonstrate that this causal semantic communication framework can be adapted to various DT applications, such as autonomous systems, smart manufacturing, and predictive maintenance, where understanding causality can enhance performance and decision-making. Their work highlights the potential of causal semantics and imitation learning in creating intelligent, adaptive DTs capable of high-quality communication, especially in scenarios demanding efficient data management and enhanced situational awareness.

In [10] Tang et al. present a novel framework for UAV-assisted DT synchronization using tiny machine learning (TinyML)-based semantic communications. The study focuses on leveraging

UAVs (Unmanned Aerial Vehicles) to support efficient and reliable data exchange for DT synchronization, particularly in remote or mobile environments where traditional connectivity might be challenging. The framework integrates TinyML—optimized machine learning models suitable for resource-limited devices—to enable semantic communication between UAVs and DTs. This approach allows the system to extract meaningful, context-relevant information from raw data, reducing communication bandwidth while preserving critical insights necessary for DT operations. The UAV-assisted DT system can prioritize essential data by focusing on semantic content, improving synchronization speed and accuracy in dynamic or large-scale environments. The authors highlight that this TinyML-based semantic communication approach is especially beneficial for IoT applications where low power and efficient data processing are key requirements. The study demonstrates that UAVs equipped with TinyML can support real-time DT updates even in challenging scenarios, such as disaster response or environmental monitoring, where reliable and adaptive DT synchronization is crucial. The paper showcases the potential of combining UAVs and TinyML to advance DT capabilities, particularly for mobile and remote IoT deployments requiring high-efficiency semantic communication.

While existing works have made notable progress in optimizing semantic communication resources for HDT systems, they fall short in providing the adaptive flexibility essential for fully efficient resource allocation. HDT systems, which frequently operate in real-time and with varying application demands, experience fluctuating resource needs based on the dynamics of the monitored physical entity and environmental changes. For instance, a health-monitoring HDT application may require increased data processing and transmission in response to an identified anomaly or critical health metric, whereas stable metrics would demand fewer resources. Achieving efficient resource management in these systems calls for strategies that go beyond static and generic optimization, allowing for dynamic adjustments in response to changing application demands. Such adaptability is crucial to avoid resource wastage and ensure that communication and computational resources are aligned precisely with application needs. Without this flexibility, HDT systems face a critical limitation: they risk over-allocating resources to low-demand tasks, leading to unnecessary energy consumption, or under-allocating resources to high-demand tasks, resulting in delays or incomplete data processing. This imbalance can ultimately degrade the system’s accuracy, responsiveness, and

overall performance.

To support high performance with minimal energy use, HDT systems must intelligently scale their resource allocation, balancing responsiveness with energy efficiency. The absence of such adaptive mechanisms reveals a key gap in current approaches. For HDT systems to be both sustainable and effective, they require a resource management framework that continuously adapts to evolving application requirements, optimizing both performance and resource efficiency.

Chapter 3

Multi-Encoder Semantic

Communication for Human Digital Twin

Synchronization

This chapter introduces the multi-encoder semantic communication approach for HDT synchronization, defining the system model and formulating the associated optimization problem. It also presents the solution to this optimization problem using a genetic algorithm, detailing the solution algorithm and simulation process. Finally, the chapter presents the results and compares them with those of existing works.

3.1 System Model

We propose a multi-encoder semantic communication system tailored for HDT applications, as illustrated in Figure 3.1. This system is designed to meet the diverse demands of Physical Twin (PT) updates while optimizing power and time efficiency. To achieve this, the system comprises L semantic encoders, each with a distinct encoding accuracy level. The accuracy of each encoder is proportional to the frequency cycles it employs during the encoding process, which in turn influences the encoder's power consumption. This setup allows for adaptive processing, as different PT updates can be assigned to encoders with accuracy levels that best match the data's requirements.

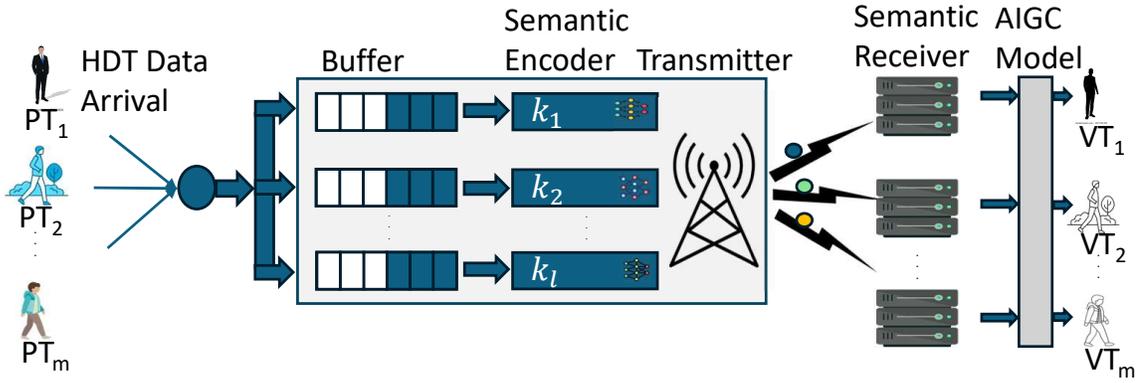


Figure 3.1: System Model

Each semantic encoder is equipped with a virtual buffer where PT updates are temporarily stored upon arrival. These updates are assigned to specific encoders based on factors such as accuracy requirements, latency requirements, and available resources. The updates within each encoder's virtual buffer are processed sequentially, allowing for efficient extraction of semantic information tailored to the demands of each HDT application. Once processed, the semantic information is transmitted over shared network channels, allowing multiple encoders to utilize common resources. This structure supports an efficient balance between high data accuracy and minimized power consumption by assigning PT updates to the most appropriate encoder based on resource constraints and synchronization requirements. This approach thus provides a flexible framework for managing the HDT system's diverse update needs while maintaining optimal resource usage.

Each Physical Twin (PT) update arrives with a unique data size and specific requirements. For each PT i , the data d_i has a minimum accuracy requirement ξ_{min}^i and a maximum latency requirement ℓ_{max}^i . PT i can be assigned to any encoder l , and this assignment is represented by the binary variable $\rho_i^l \in \{0, 1\}$, where $\rho_i^l = 1$ if PT i is assigned to encoder l ; otherwise, $\rho_i^l = 0$. The assignment of PT updates to specific encoders is based on their requirements, $(\xi_{min}^i, \ell_{max}^i)$, for data accuracy and latency, along with the encoder configuration. By aligning each PT update's requirements with the most suitable encoder, the system can optimize the allocation of resources, such as processing power and bandwidth. This alignment aims to improve overall system performance by ensuring that high-accuracy, low-latency updates are prioritized by appropriate encoders while minimizing resource usage for updates with less stringent requirements.

3.1.1 Semantic Encoder

The transformation of data d_i from a given Physical Twin PT i into meaningful semantic information, denoted as I_i^l , is achieved through a semantic encoding process. This process is captured by the expression

$$I_i^l = SE(d_i, \alpha_l(k_l)), \quad (1)$$

where SE represents a machine learning-based semantic encoding function. The function SE is designed to extract the essential, contextually relevant features from d_i rather than processing the entire raw data. This encoding approach reduces the communication burden by only transmitting information that is semantically important. In this framework α_l is a set of parameters that defines the operational settings of the encoder l . k_l represents the average semantic symbol that encoder l uses to encode each data unit, referred to here as δ . For instance, in a text-based semantic encoder, a data unit δ might correspond to a word in a sentence, capturing the meaning of each word in a compact form, as supported by prior studies [35] [36].

Each encoder may vary in terms of the value of k_l , which results in different levels of encoding accuracy, CPU frequency allocation, and power consumption. This variation among encoders allows for flexibility in how data is processed—more critical or time-sensitive data can be handled by encoders configured for high accuracy and rapid processing, while less critical data may be assigned to encoders with lower resource usage. The encoding rate μ_l^E for each encoder l is then defined as:

$$\mu_l^E = \frac{f_l}{\eta_l}, \quad (2)$$

where f_l is the CPU frequency allocation dedicated to encoder l . A higher frequency allocation indicates that encoder l can process data more quickly, resulting in a higher encoding rate. η_l is the conversion constant specific to encoder l . This constant reflects the CPU frequency required to process a single data unit δ into the corresponding semantic information unit I_δ . The parameter η_l is itself a function of encoding accuracy, which is influenced by k_l . Higher accuracy generally requires more computational resources, meaning that encoders with higher k_l values (corresponding to more

complex or precise representations) demand higher CPU frequencies to maintain the encoding rate μ_l . To manage the computational resources effectively, the system imposes a constraint on the total CPU frequency available, specifically

$$\sum_{l=1}^L f_l = f_{cpu}, \quad (3)$$

where f_{cpu} represents the total CPU frequency allocated for encoding tasks during each time slot. This constraint ensures that the total CPU frequency used by all encoders does not exceed the available processing power, enabling the system to balance resource allocation across different encoding tasks.

3.1.2 Transmission Model

The encoded semantic information is transmitted to the resultant Virtual Twin (VT) through shared channel bandwidth. At each time frame, the channels are orthogonally allocated to the encoders to eliminate intra-encoder interference [37]. The signal-to-noise ratio (SNR) between the encoder transmitter and the corresponding VT can be denoted as:

$$\gamma_i^l = \frac{P_i^{T_l} g_i h_i}{B_l N_0}, \quad (4)$$

where $P_i^{T_l}$ is the transmit power allocated to encoder l , g_i is the large-scale fading, and h_i is the small-scale Rayleigh fading coefficient. B_l is the bandwidth allocated to encoder l , and N_0 is the power spectral density. Let B_{max} be the total available bandwidth. To ensure that the bandwidth resources are managed within the system's capacity, we have the following total bandwidth constraint:

$$\sum_{l=1}^L B_l = B_{max}. \quad (5)$$

Given the encoder parameter, average semantic symbol k_l , and SNR γ_i^l , the recovered semantic information at the VT can be guaranteed an accuracy ξ_i^l , which can be expressed as

$$\xi_i^l = f(k_l, \gamma_i^l). \quad (6)$$

This accuracy function f indicates that higher k_l values or improved SNR can enhance the precision of the transmitted semantic information, which is crucial for applications requiring reliable updates in the HDT framework. According to [36], the semantic transmission rate can be defined as:

$$\mu_i^{T_l} = \frac{B_l \xi_i^l}{\Delta_l}, \quad (7)$$

where Δ_l is the per semantic unit of encoder l . On average, Δ_l can be derived as:

$$\Delta_l = \frac{k_l}{I_\delta}. \quad (8)$$

This setup reflects the interplay between encoder settings, bandwidth allocation, and semantic accuracy, highlighting the importance of dynamic resource management in HDT systems to ensure efficiency across various operational conditions.

3.2 Problem Formulation

The PT i data d_i is first expressed in terms of the data unit δ as $N_i \delta$, where N_i is the number of data units contained in the dataset. The PT i 's update data is assigned with ρ_i^l to encoder l as

$$\rho_i^l = g(d_i, \xi_{min}^i, \ell_{max}^i, k_l), \quad (9)$$

where,

$$\sum_{l=1}^L \rho_i^l = 1, \forall i \in \{1, 2, \dots, m\}. \quad (10)$$

Equation (10) ensures that each user is assigned to only one encoder. The assigned dataset joins the buffer of the allocated encoder. The buffer, which operates on a first-in-first-out protocol, leads the dataset to the semantic encoder, and the encoded semantic information is transmitted to the VT over

the network. The total latency ℓ_i^l incurred when PT i is assigned to encoder l can be given by

$$\ell_i^l = \ell_i^{E_l} + \ell_i^{T_l}, \quad (11)$$

where $\ell_i^{E_l}$ is the encoding latency and $\ell_i^{T_l}$ is the transmission latency. To synchronize the encoders so that the time frames align, we want the encoders to process the same number of PT updates over a time frame. Hence, we define the latency as:

$$\ell_i^{E_l} + \ell_i^{T_l} \leq \min\left(\frac{\tau}{n}, \ell_{max}^i\right) \quad n \in \mathbb{Z}^+, \quad (12)$$

where τ is the time frame interval, and n defines the number of PT updates each encoder processes in a time frame. We label n as the inner iteration in our simulation result. Next, we address each of the factors contributing to the latency, which includes the encoding time $\ell_i^{E_l}$ and the transmission time $\ell_i^{T_l}$.

To estimate the encoding delay $\ell_i^{E_l}$ when the PT i update data is allocated to encoder l , we express the encoding delay from (2) as

$$\ell_i^{E_l} = \frac{N_i}{\mu_l^{E_l}}. \quad (13)$$

The corresponding power consumed for the encoding of the user data, according to circuit theory, can be obtained as

$$P_i^{E_l}(t) = \kappa f_l^3, \quad (14)$$

where κ is the power coefficient that depends on the hardware architecture. $\ell_i^{T_l}$ represents the delay incurred from transmitting the semantic information I_i which can be calculated as

$$\ell_i^{T_l} = \frac{I_i}{\mu_i^{T_l}}. \quad (15)$$

Given the transmission power is $P_i^{T_l}$, the total power utilized by encoder l for user i can be expressed as

$$P_i^l = P_i^{E_l} + P_i^{T_l}. \quad (16)$$

The total power consumed for the encoding and transmission process of PT data assigned to encoder l can be defined as

$$P_l = \sum_{i=1}^m \rho_i^l P_i^l. \quad (17)$$

In this thesis, we formulate the performance evaluation and optimization of a multi-encoder semantic communication system for HDT updates. The system's performance is defined in terms of accuracy and system latency as follows:

$$\psi = \sum_{l=1}^L \sum_{i=1}^m \rho_i^l \left(\Phi \xi_i^l + \frac{\phi}{\ell_i^l} \right), \quad (18)$$

where Φ and ϕ are the accuracy and latency scale parameters. $\sum_{i=1}^m \rho_i^l \quad \forall l \in L$ represents the number of HDT updates assigned to each encoder in a time frame. The optimization variables include encoder queue assignment, computational frequency, bandwidth allocation, and power allocation, which can be concatenated as $\mathbf{A} = \{\boldsymbol{\rho}, \mathbf{f}, \mathbf{B}, \mathbf{P}\}$, where, $\boldsymbol{\rho} = \{\rho_i^l\}_{i \in m, l \in L}$, $\mathbf{f} = \{f_l\}_{l \in L}$, $\mathbf{B} = \{B_l\}_{l \in L}$, and $\mathbf{P} = \{P_l\}_{l \in L}$. Our goal is to maximize the overall system performance as formulated in the following optimization problem:

$$\mathbf{P0} \quad \max_{\mathbf{A}} \quad \psi \quad (19a)$$

$$\text{s.t.} \quad C_1 : \quad \rho_i^l \in \{0, 1\}, \quad (19b)$$

$$C_2 : \quad \sum_{l=1}^L \rho_i^l = 1, \forall i \in \{1, 2, \dots, m\}, \quad (19c)$$

$$C_3 : \quad P_l \leq \bar{P}_l, \quad (19d)$$

$$C_4 : \quad \sum_{l=1}^L B_l \leq B_{max}, \quad (19e)$$

$$C_5 : \quad \sum_{l=1}^L f_l \leq f_{cpu}, \quad (19f)$$

$$C_6 : \quad \ell_i^{E_l} + \ell_i^{T_l} \leq \min \left(\frac{\tau}{n}, \ell_{max}^i \right) \quad n \in \mathbb{Z}^+, \quad (19g)$$

$$C_7 : \quad \xi_i^l \geq \xi_{min}^i. \quad (19h)$$

where C_1 and C_2 ensure the binary allocation of a data update to only one of the encoders, C_3 is the power allocation constraint for the encoders, C_4 and C_5 indicate the maximum available CPU frequency, and transmission bandwidth for each time slot, respectively, and C_6 and C_7 indicate the latency and accuracy constraints of each PT update, respectively. Obviously, the formulated problem in (19) is a MINLP, due to its mixed-integer and nonlinear nature. The mixed integer nature is a result of the presence of both binary variables (encoder assignments) and nonlinear relationships among continuous resource allocation variables. Specifically, the semantic accuracy ξ defined as $f(k_l, \gamma_i^l)$ depends on the performance of a machine learning algorithm that interprets and conveys the meaning of transmitted data. This dependency introduces nonlinearity and complexity into the objective function and constraints. Direct computation of semantic accuracy for every possible configuration would require extensive resources due to the intricate interactions between communication parameters and machine learning interpretation models. To mitigate this computational challenge, this thesis utilizes a semantic accuracy lookup table, as suggested in prior work [36], which provides empirically validated values for ξ based on different configurations. By using the empirical values from the lookup table, we can efficiently estimate semantic accuracy across different configurations without recalculating it each time. This approach not only simplifies computations but also leverages validated empirical data to approximate semantic accuracy, thus maintaining the integrity of semantic communication requirements in the optimization process.

3.3 Genetic Algorithm Based Solution

To solve the optimization problem presented in (19), this thesis adopts the Genetic Algorithm (GA) approach, known for its robustness in navigating complex solution landscapes and avoiding local minima. This approach is particularly suitable for the challenges posed by the semantic accuracy function and the high-dimensional search space of this problem. Our GA is designed to incorporate effective encoding, fitness evaluation, genetic operators, constraint handling, and appropriate termination criteria to ensure convergence and high performance [38]. The encoding scheme includes binary encoding for ρ_i^l variables, representing each possible allocation as a binary gene, while power, bandwidth, and CPU frequency values are encoded as real-valued genes, which are

bounded by their respective constraints. This hybrid encoding scheme enables GA to explore both discrete and continuous spaces effectively. The fitness function is centered around the objective ψ , modified by penalty terms for constraint violations to produce a reward function that evaluates each solution's feasibility and quality. The reward function for the GA is expressed as follows:

$$R(\mathbf{A}) = \psi(\mathbf{A}) - P_1(\mathbf{A}) - P_2(\mathbf{A}) - P_3(\mathbf{A}) - P_4(\mathbf{A}), \quad (20)$$

where $P_1(\mathbf{A})$, $P_2(\mathbf{A})$, $P_3(\mathbf{A})$, and $P_4(\mathbf{A})$ are the penalty functions for the constraints C_2 , C_3 , C_6 , and C_7 , respectively, which can be expressed as

$$P_1(\mathbf{A}) = c_1 \left(\sum_{i=1}^m \left| \sum_{l=1}^L \rho_i^l - 1 \right| \right), \quad (21)$$

$$P_2(\mathbf{A}) = c_2 \sum_{l=1}^L \max(P_l - \bar{P}_l, 0), \quad (22)$$

$$P_3(\mathbf{A}) = c_3 \sum_{l=1}^L \sum_{i=1}^m \max \left(\ell_i^{E_l} + \ell_i^{T_l} - \min \left(\frac{\tau}{n}, \ell_{max}^i \right), 0 \right), \quad (23)$$

$$P_4(\mathbf{A}) = c_4 \sum_{l=1}^L \sum_{i=1}^m \max \left(\xi_{min}^i - \xi_i^l, 0 \right), \quad (24)$$

where c_1 , c_2 , c_3 , and c_4 are the penalty scaling constants. The values of the bandwidth and frequency are normalized to meet the constraints C_4 and C_5 , respectively. By applying this carefully designed GA-based approach, we address the MINLP problem effectively, achieving solutions that optimize system performance under mixed-integer and nonlinear constraints. The GA-based algorithm for the MINLP problem is summarized in Algorithm 1.

The algorithm begins by initializing various parameters, including the population size, simulation parameters, and the number of generations N , which defines the maximum number of iterations for the algorithm. Additionally, the channel state information and data arrival parameters are set, which represent the communication environment in which the system operates. The total number of genes, num_genes , is calculated as $mL + 3L$, where m is the number of PT updates and L is the number of encoders. The expected output is the complete optimization vector A , which includes

Algorithm 1 GA-based Solution Algorithm

- 1: **Input:** Initialize the population and simulation parameter as shown in Table 3.1.
 - 2: **Output:** Optimal solution \mathbf{A}^* and reward $R(\mathbf{A})^*$.
 - 3: $num_genes \leftarrow m * L + 3L$
 - 4: Initialize the channel state information and data arrival.
 - 5: $last_fitness \leftarrow -\infty$
 - 6: $N =$ Maximum number of generations
 - 7: $P_{init} \leftarrow$ randomly initialize population
 - 8: **while** $N > 2$ **do**
 - 9: Select parents from the population
 - 10: Perform crossover and mutation to generate $new_offsprings$
 - 11: $P_{new} \leftarrow new_offsprings$
 - 12: Evaluate the fitness of each offspring with $R(\mathbf{A})$ and find $best_fitness$
 - 13: $last_fitness \leftarrow best_fitness$ and $N \leftarrow N - 1$
 - 14: **end while**
 - 15: Return the best solution \mathbf{A}^* and its fitness reward $R(\mathbf{A})^*$.
-

binary allocation variables ρ_i^l and continuous variables for power, bandwidth, and CPU frequency. In Step 7, the initial population P_{init} is randomly generated. This population consists of individuals, where each individual represents a possible solution to the optimization problem. The individuals include both binary genes for encoder assignments and real-valued genes for power, bandwidth, and CPU frequency. The initial random generation ensures that the search space is explored from the start, offering a diverse set of solutions. The core of the algorithm operates within a loop that runs for a predefined number of generations, as specified in Step 8. In each generation, steps 9, 10, 11, 12, and 13 are performed. In Step 9, parents are selected from the current population based on their fitness. Higher-fitness individuals are more likely to be selected, ensuring that the genetic material of good solutions has a higher chance of propagating into the next generation. In Step 10, the selected parents undergo crossover and mutation to create a new set of offspring. Crossover combines the genetic material of the parents to produce new solutions, mixing their characteristics. In Step 11, the fitness of each offspring is evaluated using the fitness function defined in (20). The fitness function includes the core objective ψ and penalty terms for constraint violations. These penalties account for the allocation, power, latency, and accuracy constraints. The fitness value indicates how well the offspring satisfies the optimization problem, with higher values corresponding to better solutions. In Step 12, the best solution (i.e., the individual with the highest fitness score) is identified from the current generation. The fitness score is updated in the variable $last_fitness$, and the loop

continues by decrementing N , which tracks the number of generations remaining. The loop continues until N reaches 2, which serves as the termination condition. Once the maximum number of generations has been reached, the algorithm terminates. In Step 13, the algorithm returns the best solution \mathbf{A}^* found during the process, along with its associated fitness reward $R(\mathbf{A})^*$. This solution contains the optimal allocation of PT updates to encoders and the optimal continuous values for power, bandwidth, and CPU frequency that maximize system performance while satisfying the constraints outlined in the optimization problem. This approach allows the GA to explore a large solution space efficiently while ensuring that solutions improve over time, gradually converging to the optimal solution. The algorithm is designed to handle the complexity and nonlinearity of the MINLP problem by employing a combination of genetic operators and penalty functions to maintain feasible solutions. The final output provides an optimal action plan for encoder allocation and resource allocation that maximizes system performance while adhering to all constraints.

3.4 Simulation Results

To demonstrate the effectiveness of the proposed multi-encoder semantic system for HDT synchronization, this section presents the simulation parameters used in the GA-based optimization and then proceeds to compare the performance of the proposed system with existing semantic encoder frameworks. As shown below, Table 3.1 presents the key simulation parameters for both the GA and the multi-encoder semantic system. These values were carefully selected based on previous research in semantic communication and GA, such as those discussed in [36] and [38] respectively. The GA-based parameters include settings for the crossover and mutation probabilities, the number of generations, population size, and the types of crossover and mutation used. These parameters define how the GA will explore and optimize the search space for the best solution to the resource allocation problem.

The implementation environment for the algorithm involves Python 3 as the programming language, leveraging its simplicity and wide range of libraries. The IDE used is PyCharm, specifically the Professional Edition, which offers features like intelligent code suggestions, debugging tools, virtual environment management, and an integrated terminal. The operating system is Windows 10.

Table 3.1: Simulation Parameters

Genetic Algorithm	
Crossover probability	0.4
Mutation probability	0.08
Number of generations	300
Number of genes	129
Population size	50
Mutation type	Random
Crossover type	Single point
Multi-encoder Semantic	
Number of PT updates m	40
Number of Encoders L	3
Power constant κ	$10^{-11}WHz^{-3}$
CPU frequency f_{cpu}	10^5Hz
Total Bandwidth B	10^7Hz
Phi Φ	1100
phi ϕ	0.9
power threshold \bar{P}_l	$3KW$

For a fair comparison between our proposed multi-encoder semantic model and the existing semantic structure, we implemented three additional semantic encoder frameworks, each configured with identical k_l values. Although the existing system uses a single-encoder framework, we adapted it to replicate the encoding rate achievable in the proposed multi-encoder model. We then applied the genetic algorithm to optimize resource allocation across all models. In these comparisons, our proposed model is labeled $k_l = [9, 6, 3]$, while the benchmark models are configured with $k_l = [9, 9, 9]$, $k_l = [6, 6, 6]$, and $k_l = [3, 3, 3]$. Typically each of the benchmark setups represents a single encoder with $k_l = 9$, $k_l = 6$, and $k_l = 3$ respectively. By evaluating the performance of these models, the simulation results will reveal how the multi-encoder system outperforms the single-encoder configurations, showing its ability to achieve higher accuracy, lower latency, and overall better system performance in the context of semantic communication for HDT synchronization.

Figure 3.2 illustrates the performance of the proposed multi-encoder semantic system in terms of latency violations. Latency violation refers to the extent to which the actual latency exceeds the predetermined threshold, providing an indication of how much delay surpasses acceptable limits. The figure clearly highlights that our proposed system performs better than the benchmark models,

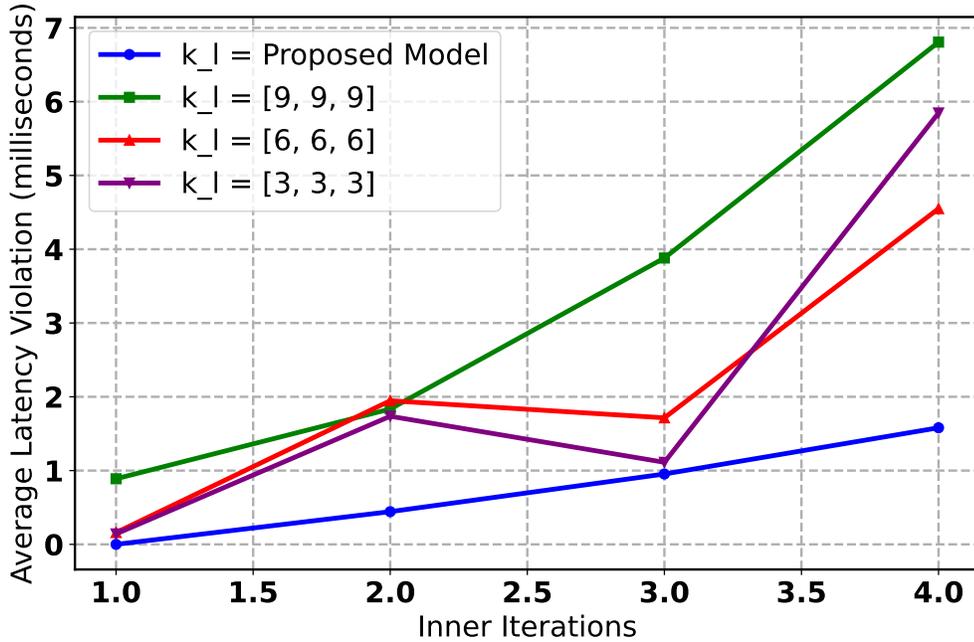


Figure 3.2: The Latency Violation versus the number of inner iterations n

particularly in reducing latency violations. As we increase the number of internal iterations, the system faces stricter resource constraints, which leads to a noticeable rise in latency violations. This trend is expected because higher internal iterations imply a more complex resource allocation scenario, putting more strain on the system. However, the proposed multi-encoder system effectively mitigates this issue by dynamically adjusting the resource allocation based on the priority of different updates. By prioritizing more sensitive updates (those that are crucial for maintaining semantic accuracy) over less sensitive ones, the system optimizes its resources in real-time, ensuring that critical updates are processed with lower latency. This adaptive resource allocation strategy is key to the system’s ability to minimize latency violations, even when the resources are constrained. The figure demonstrates the robustness and efficiency of the proposed model, which not only maximizes system performance but also ensures that latency remains within acceptable limits despite varying resource conditions.

The results in Figure 3.3 highlight the importance of selecting adequate k_l values to maintain accuracy in updates. The configuration where $k_l = [3, 3, 3]$ demonstrates a clear inability to meet accuracy requirements as the number of inner iterations increases. In contrast, configurations with

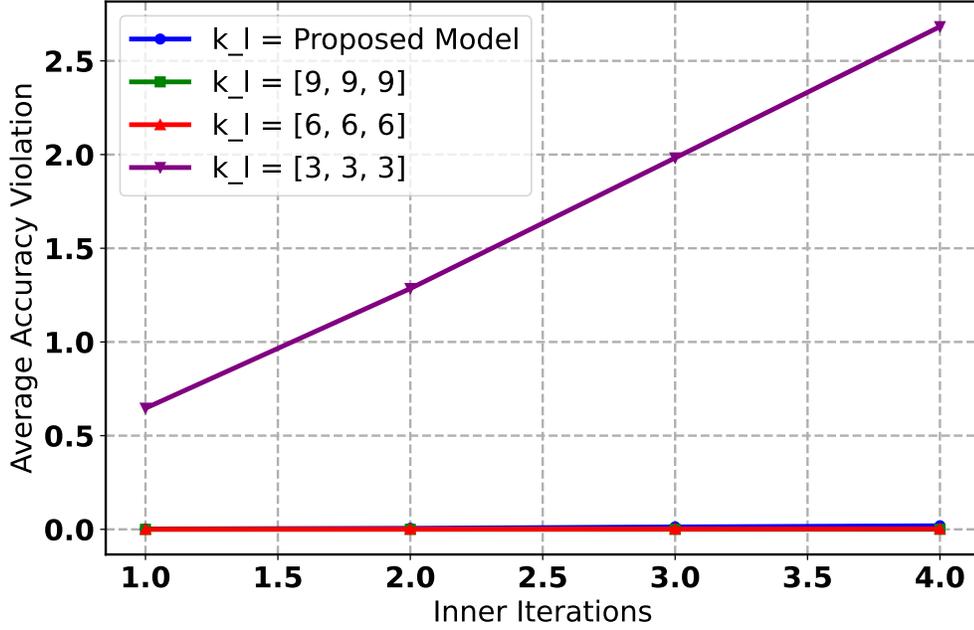


Figure 3.3: The Accuracy Violation versus the number of inner iterations n

higher k_l values, such as $k_l = [9, 9, 9]$, $k_l = [6, 6, 6]$, and the proposed $k_l = [9, 6, 3]$, consistently avoid accuracy violations. This reinforces the proposed model’s ability to meet accuracy demands by appropriately allocating resources to satisfy the update requirements. The results underscore the significance of selecting and adjusting k_l values to achieve the right trade-off between resource allocation and the accuracy of updates. By allocating resources strategically, the proposed model ensures that the system can meet accuracy requirements, effectively avoiding violations and maintaining optimal system performance under varying conditions.

Figure 3.4 illustrates the power consumption trends for the different configurations as the number of iterations increases. As expected, power consumption increases with higher k_l values, reflecting the greater computational demands and higher frequency utilization required for operation. Specifically, configurations with larger values, such as $k_l = [9, 9, 9]$ and $k_l = [6, 6, 6]$, naturally exhibit higher power consumption due to the increased encoding and processing capacity needed. Although the proposed model with $k_l = [9, 6, 3]$ consumes more power than the configuration with $k_l = [3, 3, 3]$ the additional power consumption is justified by the substantial performance improvements it offers in terms of accuracy and latency violation. The dynamic resource allocation strategy

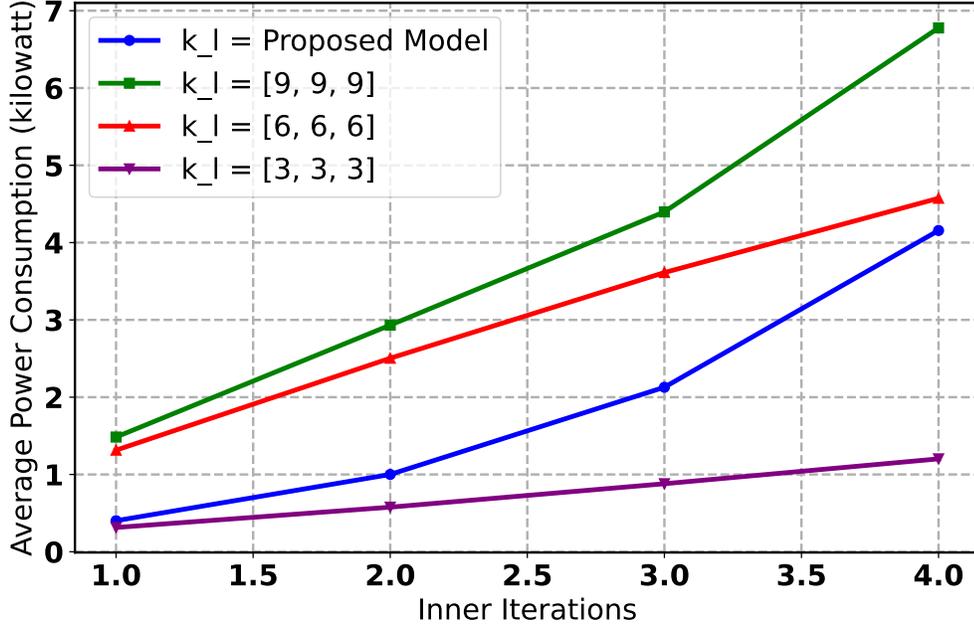


Figure 3.4: The Power Consumption versus the number of inner iterations n

of the proposed model ensures that power is utilized more efficiently, prioritizing critical updates and optimizing performance under constrained resources. In this context, the marginal increase in power consumption is a trade-off for better accuracy and reduced latency violations, demonstrating that the benefits of the proposed system outweigh the costs associated with higher power requirements. This result highlights the importance of considering both power efficiency and system performance when designing semantic communication systems. Although power consumption is a critical factor, the proposed model’s ability to meet accuracy and latency constraints makes the higher power consumption a reasonable and valuable investment.

Figure 3.5, presents the reward metric as defined in (20), offering insights into how different configurations perform with respect to both penalties and performance gains across the inner iterations. The results indicate that the proposed model, along with configurations that utilize higher k_l values, such as $k_l = [9, 9, 9]$ and $k_l = [6, 6, 6]$ consistently maintain a stable average total reward throughout the iterations. This stability is a strong indicator that these configurations are able to balance the various performance factors, effectively minimizing penalties and offsetting any incurred penalties with substantial performance gains in areas like accuracy and latency. For example, the

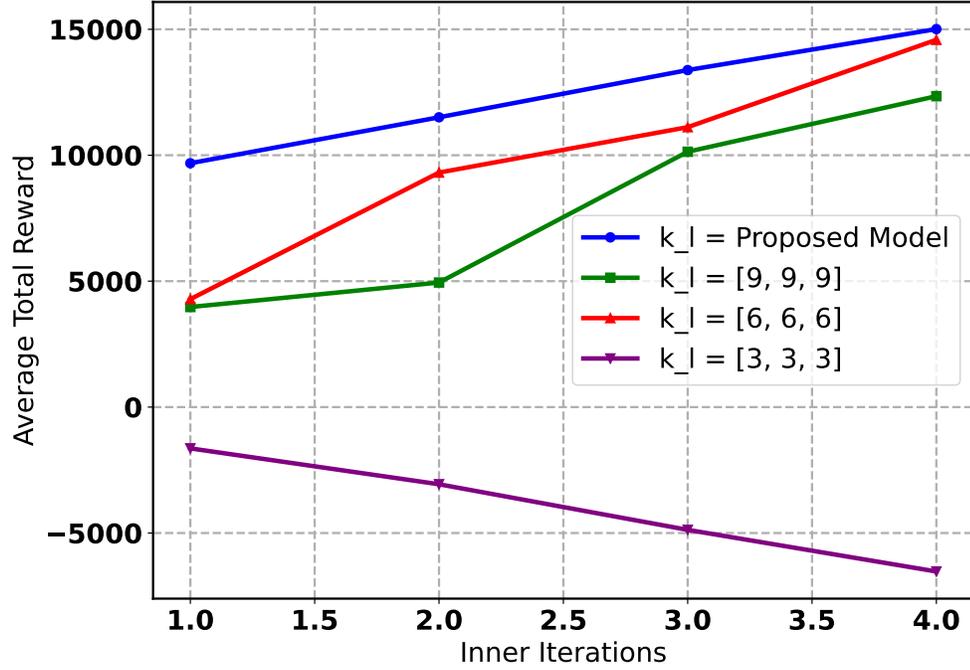


Figure 3.5: The Average Total Reward R versus the number of inner iterations n

configuration with $k_l = [9, 9, 9]$ is expected to achieve high accuracy in the updates, as reflected by its reward. However, as shown in Figure 3.2 and Figure 3.4, it does face penalties related to latency and power constraints. Despite these penalties, the significant accuracy reward, which has a high coefficient in the reward function, compensates for the incurred losses. This demonstrates how higher k_l values can enhance overall system performance, even when some trade-offs are made with respect to power consumption and latency violations. On the other hand, the configuration with $k_l = [3, 3, 3]$ faces a different challenge. While it experiences a smaller penalty related to power consumption, it incurs a much higher accuracy penalty. This is due to its inability to meet the accuracy requirements for critical PT updates, which are vital for the overall performance of the system. The higher accuracy penalty negatively impacts the reward, as the lack of sufficient semantic accuracy outweighs any minor power savings, underscoring the importance of selecting appropriate k_l values that meet both accuracy and latency constraints. Thus, the reward metric highlights the trade-offs between accuracy, latency, and power consumption, emphasizing the proposed model's capability to balance these factors more effectively than other configurations.

Chapter 4

Lyapunov-Assisted DRL-based Multi-Encoder Semantic Communication for HDT Synchronization

To explore the system model stability over a long-term, we re-formulate the short-term problem in chapter 3 to a long-term optimization problem. This is necessary to balance the immediate performance with future goals, such as maintaining queue stability, ensuring sustainable energy usage, and achieving high throughput over the entire network's operation. This will mitigate the risk of short-sighted decisions prioritizing immediate gains at the expense of long-term efficiency and resilience. It integrates cumulative effects and evolving system states, ensuring that the model operates optimally not just in the short term but also over an extended horizon. Thus, the reformulated problem provides a robust framework for addressing both present and future demands, contributing to the overall stability and sustainability of the system.

4.1 System Model

To extend the system model and formulations introduced in Chapter 3 to a long-term scenario, we define the arrival pattern and queue management for PT updates and examine their impact on system performance, particularly the latency.

The PT update arrival model assumes updates arrive independently over time, each with unique attributes: data size $d_i(t)$, minimum accuracy requirement $\xi_{\min}^i(t)$, and maximum latency requirement $\ell_{\max}^i(t)$ for $i = 1, \dots, m$. Arrivals follow a Poisson process, and each attribute is modeled as independent and identically distributed (i.i.d.) across time slots. Specifically, the data size $d_i(t)$ is sampled from a uniform distribution $U(d_{\text{low}}, d_{\text{high}})$, the accuracy requirement $\xi_{\min}^i(t)$ from $U(\xi_{\text{low}}, \xi_{\text{high}})$, and the latency requirement $\ell_{\max}^i(t)$ from $U(\ell_{\text{low}}, \ell_{\text{high}})$.

4.1.1 Queuing Model

Assume an arrival rate $\lambda(t)$, where each update in the arrival stream is assigned to a virtual queue Q_l that is serviced by encoder l . Let $\lambda_l(t)$ represent the portion of updates allocated to encoder l at time t , such that

$$\sum_{l=1}^L \lambda_l(t) = \lambda(t). \quad (25)$$

The queue length of encoder l at the beginning of time slot t is denoted by $Q_l(t)$, and its state update at $t + 1$ is defined as

$$Q_l(t + 1) = \max(Q_l(t) - \mu_l(t), 0) + \lambda_l(t), \quad (26)$$

where $\mu_l(t)$ represents the encoding rate of encoder l at time t . To satisfy causality, we impose the constraint $\mu_l(t) \leq Q_l(t)$, ensuring that $Q_l(t) \geq 0$ for all t . Thus, we can simplify the queue update equation to

$$Q_l(t + 1) = Q_l(t) - \mu_l(t) + \lambda_l(t). \quad (27)$$

We assume that the queue buffer capacity is sufficiently large to prevent data loss. However, controlling the average waiting time is crucial for meeting PT requirements, as the average waiting time is directly proportional to the average queue length. Therefore, maintaining a stable queue length is essential to ensure finite processing times. According to [39], queue stability is achieved under the

following constraint:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}\{Q_l(t)\} < \infty, \forall l \in \{1, 2, \dots, L\}. \quad (28)$$

This condition ensures that the expected queue length remains bounded over time, supporting stable processing across all queues.

4.2 Problem Formulation

Each PT i arrives with $d_i(t)$, $\xi_{\min}^i(t)$, $\ell_{\max}^i(t)$. The data $d_i(t)$ can be expressed in terms of the data unit δ as $N_i(t)\delta$, where $N_i(t)$ is the number of data units contained in the dataset. The PT i update data is assigned with $\rho_i^l(t)$ to encoder l :

$$\rho_i^l(t) = g(d_i(t), \xi_{\min}^i(t), \ell_{\max}^i(t), Q_l(t), k_l). \quad (29)$$

The assigned dataset joins the queue of the allocated encoder. The queue which operates on a first-in-first-out (FIFO) protocol, leads the dataset to the semantic encoder. The encoded semantic information is then transmitted to the VT over the network. The total latency ℓ_i^l incurred when PT i is assigned to encoder l can be given by

$$\ell_i^l(t) = \ell_i^{Q_l}(t) + \ell_i^{E_l}(t) + \ell_i^{T_l}(t), \quad (30)$$

where $\ell_i^{Q_l}(t)$ is the queue delay, $\ell_i^{E_l}$ is the encoding latency, and $\ell_i^{T_l}$ is the transmission latency. To synchronize the encoders such that the time frame aligns, we want the encoders to process the same number of PT updates over a time frame. Hence, we define the latency as:

$$\ell_i^{E_l}(t) + \ell_i^{T_l}(t) \leq \min\left(\frac{\tau}{n}, \ell_{\max}^i(t) - \ell_i^{Q_l}(t)\right), \quad n \in \mathbb{Z}^+, \quad (31)$$

where τ is the time frame interval, and n defines the number of PT updates each encoder processes in a time slot. Next, we address each of the factors contributing to the latency, which include the queue waiting time $\ell_i^{Q_l}(t)$, the encoding time $\ell_i^{E_l}(t)$, and the transmission time $\ell_i^{T_l}(t)$. The encoding and transmission time follows the definition in (13) and (15) respectively. To evaluate the waiting

time on queue $Q_l(t)$, the average queue length can be expressed as

$$Q_l = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T Q_l(t). \quad (32)$$

From Little's law $L = \lambda W$, where W represents the waiting time in the queue, λ is the average arrival rate, and L is the queue length. The waiting time on $Q_l(t)$ can therefore be defined as:

$$\ell_i^{Q_l}(t) = \lim_{T \rightarrow \infty} \frac{1}{\lambda_l T} \sum_{t=1}^T Q_l(t). \quad (33)$$

From (2) and (7), the encoding-transmission rate for a single PT update i over the encoding and transmission resources l is expressed as

$$\mu_i^l(t) = \frac{\mu_l^E(t) \mu_i^{T_l}(t)}{\mu_l^E(t) + \mu_i^{T_l}(t)}. \quad (34)$$

The total encoding-transmission rate of encoder l can therefore be derived as

$$\mu_l(t) = \sum_{i=1}^m \rho_i^l(t) \mu_i^l(t). \quad (35)$$

The system performance at time t in terms of accuracy and system latency can be defined as:

$$\psi(t) = \sum_{l=1}^L \sum_{i=1}^m \rho_i^l(t) \left(\Phi \xi_i^l(t) + \frac{\phi}{\ell_i^l(t)} \right) \quad (36)$$

where Φ and ϕ are the accuracy and latency scale parameters. $\sum_{i=1}^m \rho_i^l(t) \quad \forall l \in L$ represents the number of HDT updates each encoder processes in a time frame. The optimization variables include encoder queue assignment, computational frequency, bandwidth allocation, and power allocation. The optimization variable can be concatenated as $\mathbf{A} = \{\mathbf{A}(t)\}_{t \in T}$, where $\mathbf{A}(t) = \{\boldsymbol{\rho}(t), \mathbf{f}(t), \mathbf{B}(t), \mathbf{P}(t)\}_{t \in T}$ comprehensively. Specifically, $\boldsymbol{\rho}(t) = \{\rho_i^l(t)\}_{i \in N, l \in L}$, $\mathbf{f}(t) = \{f_l(t)\}_{l \in L}$, $\mathbf{B}(t) = \{B_l(t)\}_{l \in L}$, and $\mathbf{P}(t) = \{P_l(t)\}_{l \in L}$.

Our goal is to maximize the overall system performance over a long-term as formulated in the

following optimization problem:

$$\mathbf{P0} \quad \max_{\mathbf{A}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[\psi(t)] \quad (37a)$$

$$\text{s.t. } C_1 : \quad \rho_i^l(t) \in \{0, 1\}, \quad (37b)$$

$$C_2 : \quad \sum_{l=1}^L \rho_i^l(t) = 1, \forall i \in \{1, 2, \dots, m\}, \quad (37c)$$

$$C_3 : \quad \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T P_l(t) \leq \bar{P}_l, \quad (37d)$$

$$C_4 : \quad \sum_{l=1}^L B_l(t) \leq B_{max}, \quad (37e)$$

$$C_5 : \quad \sum_{l=1}^L f_l(t) \leq f_{cpu}, \quad (37f)$$

$$C_6 : \quad \ell_i^{E_l}(t) + \ell_i^{T_l}(t) \leq \min\left(\frac{\tau}{n}, \ell_{max}^i(t) - \ell_i^{Q_l}(t)\right) \quad n \in \mathbb{Z}^+, \quad (37g)$$

$$C_7 : \quad \xi_i^l(t) \geq \xi_{min}^i(t), \quad (37h)$$

$$C_8 : \quad \mu_l(t) \leq Q_l(t), \quad (37i)$$

$$C_9 : \quad \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}\{Q_l(t)\} < \infty, \forall l \in \{1, 2, \dots, L\}. \quad (37j)$$

where C_1 and C_2 ensure binary allocation of a data update to only one of the encoders, C_3 is the long-term power allocation constraint for the encoders. C_4 and C_5 indicate the maximum available CPU frequency and transmission bandwidth for each time slot. C_6 and C_7 represent the latency and accuracy constraints for each PT update. C_8 ensures that the encoding-transmission rate at every time frame does not exceed the available data in the queue. C_9 ensures the queue stability.

The problem **P0** represents a multistage stochastic optimization problem with random channel conditions and data arrivals. It involves jointly determining the queue allocation and optimizing resource allocation in a sequential time frame under these stochastic conditions. The decision variables in **P0** are interdependent, with resource management decisions made at one time step affecting the system's performance in future time steps. This complexity, along with the dynamic nature of

the problem, suggests that traditional optimization approaches may struggle to find effective solutions.

One approach to address this problem is to apply Deep Reinforcement Learning (DRL), which has been shown to achieve optimal solutions for Markov Decision Process (MDP) problems. While DRL is a powerful tool for complex sequential decision-making, it faces significant challenges in handling long-term constraints, especially when these constraints are not explicitly included in the state representation. Specifically, DRL encounters difficulties with:

- **Constraint embedding:** Incorporating constraints directly into the reward function often requires careful tuning, which may not guarantee the satisfaction of long-term constraints.
- **Accumulating long-term constraint effects:** DRL lacks an inherent mechanism to accumulate the effects of long-term constraints that are not represented in the state, which can lead to constraint violations over time.

On the other hand, Lyapunov optimization provides a mathematically rigorous approach to managing long-term constraints by transforming them into virtual queue stability problems. These virtual queues act as a persistent memory for past constraint violations, ensuring that long-term constraints are tracked and maintained over time. While Lyapunov optimization excels in managing long-term constraints, DRL is particularly well-suited to solving the underlying MDP and optimizing the utility function.

Therefore, a hybrid approach combining Lyapunov optimization with DRL is particularly well-suited for our problem, as it ensures both theoretical rigor and practical effectiveness. Specifically:

- (1) **Lyapunov optimization** is deployed to decouple the long-term stochastic problem into a series of deterministic per-frame MDP problems. This guarantees that all long-term constraints are satisfied by solving each time-frame subproblem sequentially [40].
- (2) **DRL** is employed with a model-free actor and a model-based critic to solve the per-frame MDP efficiently, leveraging DRL's strengths in optimizing utility under complex, dynamic conditions.

This combined approach addresses the limitations of DRL in handling long-term constraints while exploiting its ability to optimize sequential decision-making effectively.

4.2.1 Lyapunov Guided Problem Transformation

In this section, we deploy Lyapunov optimization to decouple the stochastic multistage problem into a per-time slot deterministic problem. First, to cope with the long-term power constraint in (37d), we introduce virtual power deficit queues, as described below:

$$Z_l(t+1) = \max [Z_l(t) + P_l(t) - \bar{P}_l, 0], \quad (38)$$

where $Z_l(0) = 0$. The virtual power queue $Z_l(t)$ represents the deviation between the current power consumption and the time-averaged power constraint in (38), indicating the degree to which the constraints are being satisfied.

To jointly control the power and the PT arrival queue, we introduce $\mathbf{Y}(t) = \{\mathbf{Z}(t), \mathbf{Q}(t)\}$ as the total queue backlog, where $\mathbf{Q}(t) = \{Q_l(t)\}_{l \in L}$ and $\mathbf{Z}(t) = \{Z_l(t)\}_{l \in L}$. The Lyapunov function can be expressed as

$$\mathbb{L}(\mathbf{Y}(t)) = \frac{1}{2} \sum_{l \in L} (Y_l(t))^2. \quad (39)$$

The corresponding Lyapunov drift can be defined as

$$\Delta \mathbb{L}(\mathbf{Y}(t)) = \mathbb{E}[\mathbb{L}(\mathbf{Y}(t+1)) - \mathbb{L}(\mathbf{Y}(t)) | \mathbf{Y}(t)]. \quad (40)$$

To maximize the long-term system performance while ensuring the power consumption and data virtual queue constraint, we define a Lyapunov-drift-plus-penalty function as

$$\Delta_V \mathbb{L}(\mathbf{Y}(t)) = \Delta \mathbb{L}(\mathbf{Y}(t)) - V \mathbb{E}[\psi(t) | \mathbf{Y}(t)], \quad (41)$$

where V is a positive value representing the trade-off between the virtual queue stability and the system performance. The upper bound of the drift is derived using the following lemma.

Lemma 1. *The Lyapunov-drift-plus-penalty defined in (41) is bounded by:*

$$\begin{aligned} \Delta_V \mathbb{L}(\mathbf{Y}(t)) &\leq G - V \mathbb{E}[\psi(t)|\mathbf{Y}(t)] + \sum_{l \in L} \mathbb{E}[Z_l(t)(P_l(t) - \bar{P}_l)|\mathbf{Y}(t)] \\ &\quad + \sum_{l \in L} \mathbb{E}[Q_l(t)(\lambda_l(t) - \mu_l(t))|\mathbf{Y}(t)]. \end{aligned} \quad (42)$$

Proof. from (38) we have

$$\begin{cases} (Z_l(t+1))^2 = Z_l(t)^2 + 2Z_l(t)(P_l(t) - \bar{P}_l) + (P_l(t) - \bar{P}_l)^2, \\ (Q_l(t+1))^2 = Q_l(t)^2 + 2Q_l(t)(\lambda_l(t) - \mu_l(t)) + (\lambda_l(t) - \mu_l(t))^2. \end{cases} \quad (43)$$

By taking the sum over L queues on both sides, we have

$$\begin{cases} \frac{1}{2} \sum_{l \in L} (Z_l(t+1))^2 - \frac{1}{2} \sum_{l \in L} (Z_l(t))^2 = \sum_{l \in L} Z_l(t)(P_l(t) - \bar{P}_l) + \\ \quad \frac{1}{2} \sum_{l \in L} (P_l(t) - \bar{P}_l)^2, \\ \frac{1}{2} \sum_{l \in L} (Q_l(t+1))^2 - \frac{1}{2} \sum_{l \in L} (Q_l(t))^2 = \sum_{l \in L} Q_l(t)(\lambda_l(t) - \mu_l(t)) + \\ \quad \frac{1}{2} \sum_{l \in L} (\lambda_l(t) - \mu_l(t))^2. \end{cases} \quad (44)$$

Taking the conditional expectation of (44) we have

$$\begin{cases} \frac{1}{2} \sum_{l \in L} \mathbb{E}[(Z_l(t+1))^2|\mathbf{Y}(t)] - \frac{1}{2} \sum_{l \in L} \mathbb{E}[(Z_l(t))^2|\mathbf{Y}(t)] \leq \\ \quad G_1 + \sum_{l \in L} \mathbb{E}[Z_l(t)(P_l(t) - \bar{P}_l)|\mathbf{Y}(t)], \\ \frac{1}{2} \sum_{l \in L} \mathbb{E}[(Q_l(t+1))^2|\mathbf{Y}(t)] - \frac{1}{2} \sum_{l \in L} \mathbb{E}[(Q_l(t))^2|\mathbf{Y}(t)] \leq \\ \quad G_2 + \sum_{l \in L} \mathbb{E}[Q_l(t)(\lambda_l(t) - \mu_l(t))|\mathbf{Y}(t)]. \end{cases} \quad (45)$$

where

$$\begin{cases} G_1 = \frac{1}{2} \sum_{l \in L} \mathbb{E}[(P_{max}^l)^2 + (\bar{P}_l)^2], \\ G_2 = \frac{1}{2} \sum_{l \in L} \mathbb{E}[(\lambda_{max}^l)^2 + (\mu_l(t))^2]. \end{cases} \quad (46)$$

where P_{max}^l and λ_{max}^l are, respectively, the maximum power consumption and the maximum PT update allocation for encoder l in a time slot. where $G = G_1 + G_2$, the sum of (45) gives

$$\begin{aligned} \Delta \mathbb{L}(\mathbf{Y}(t)) &\leq G + \sum_{l \in L} \mathbb{E}[Z_l(t)(P_l(t) - \bar{P}_l) | \mathbf{Y}(t)] \\ &\quad + \sum_{l \in L} \mathbb{E}[Q_l(t)(\lambda_l(t) - \mu_l(t)) | \mathbf{Y}(t)]. \end{aligned} \quad (47)$$

We obtain (42) by subtracting $V\mathbb{E}[\psi(t) | \mathbf{Y}(t)]$ to both sides of (47) to complete the proof. \square

The long-term multi-stage optimization problem **P0** is transformed into several deterministic problems with per-slot constraints. In each time slot, we use an opportunistic expectation minimization technique[40]. We observe the queue backlogs $\mathbf{Y}(t)$ and decide the power allocation and queue assignment control actions accordingly to minimize the upper bound in (42). With this approach, we can reformulate the optimization problem **P0** into a per-slot resource allocation problem as

$$\begin{aligned} \mathbf{P1} \quad & \min_{\mathbf{A}(t)} -V[\psi(t)] + \sum_{l \in L} [Z_l(t)(P_l(t) - \bar{P}_l) | \mathbf{Y}(t)] + \sum_{l \in L} [Q_l(t)(\lambda_l(t) - \mu_l(t)) | \mathbf{Y}(t)] \quad (48) \\ \text{s.t.} \quad & C_1, C_2, C_4 - C_8. \end{aligned}$$

In the next section, we will demonstrate how we can satisfy the long-term constraint in **P0** by implementing a low-complexity algorithm leveraging DRL. We combine a model-free DRL approach with a model-based online optimization algorithm.

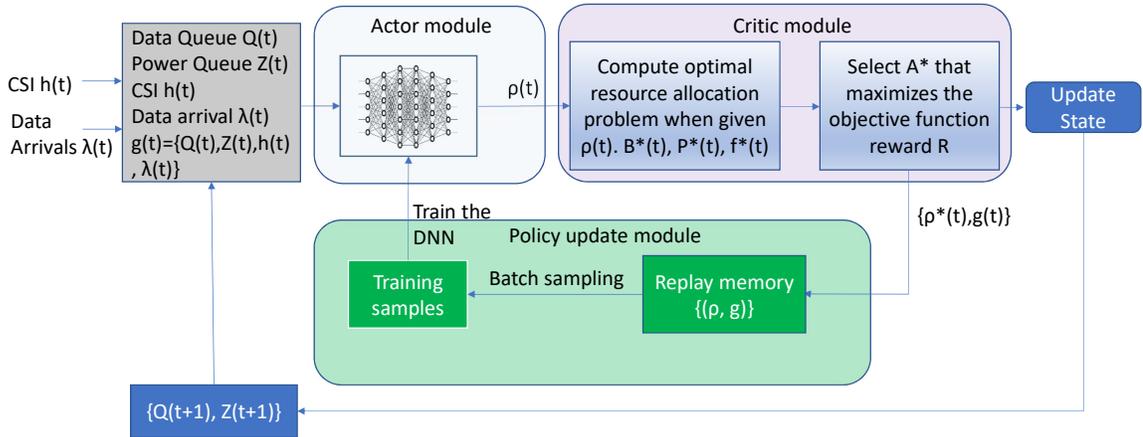


Figure 4.1: DRL framework schematic

4.3 Deep Reinforcement Learning based Online HDT Synchronization

P1 can be categorized as a Markov process because it involves sequential decision-making over time, where actions taken in each state influence future states. The variable $\mathbf{A}(t)$ consists of both binary and continuous components, which can be represented as $\mathbf{A}(t) = \{\rho(t), \mathbf{D}(t)\}$, where $\rho(t)$ denotes the binary data allocation variable, and $\mathbf{D}(t) = \{f(t), \mathbf{B}(t), \mathbf{P}(t)\}$ represents the continuous resource optimization variable set. Therefore, to solve the Markov problem involving optimal binary data allocation and continuous resource optimization, we employ a DRL-based synchronization framework, as shown in Figure 4.1. This framework consists of three key modules: a model-free actor module, a model-based critic module, and a DNN model policy update module. Given the information on queue state, data arrival, and channel state, the actor module outputs a set of data allocation actions. The critic module then evaluates and optimizes resource allocation based on these data allocations. The optimal actions and resource allocation are selected to update the queue state. Over time, the DNN module updates to improve the actor's policy. This process runs iteratively, as detailed below.

4.3.1 Model Free Actor Module

The model-free actor module is a DNN-based model that derives the optimal HDT update allocation decisions given the current queue state, channel state information, and HDT update arrival as input. It consists of a DNN, a row-wise softmax operation, and a row-wise one-hot encoding.

Given m new HDT updates arriving and L sets of encoders, the state input is reshaped and concatenated into a one-dimensional feature vector $g \in \mathbb{R}^n$, where n is the dimensionality of the input state:

$$g = [g_1, g_2, \dots, g_n] \in \mathbb{R}^n. \quad (49)$$

The neural network parameter $\theta(t)$, which is initialized randomly with a standard normal distribution at $t = 1$, maps the input g to an output matrix O , where $O \in \mathbb{R}^{m \times L}$, with m representing the number of decision dimensions and L representing the number of possible actions in each dimension:

$$O = f_{\theta(t)}(g) \in \mathbb{R}^{m \times L}, \quad (50)$$

where $f_{\theta(t)}$ represents the deep neural network. Note that the matrix O is the raw output of the network before any normalization. The raw output matrix O is then converted into a probability matrix Pr using the softmax function applied row-wise. Each row in Pr represents a probability distribution over L actions for each decision dimension:

$$Pr_{i,j} = \frac{\exp(O_{i,j})}{\sum_{k=1}^L \exp(O_{i,k})} \quad \forall i = 1, \dots, m, j = 1, \dots, L, \quad (51)$$

where $Pr_{i,j}$ is the probability of selecting action j in row (decision dimension) i . For each row i of the probability matrix Pr , we select the top k actions with the highest probabilities. Let \mathcal{T}_i be the set of indices of the top k actions for row i . We have

$$\mathcal{T}_i = \{a_i^{(1)}, a_i^{(2)}, \dots, a_i^{(k)}\} = \arg \max_j Pr_{i,j} \quad \forall i = 1, \dots, m. \quad (52)$$

To satisfy the constraint specified in (37b) and (37c), for each selected action $a_i^{(k)}$, we create a one-hot encoded matrix $M_{i,j}^{(k)}$, where each row i contains a single 1 at the position corresponding to the

highest probability action and 0 elsewhere. Mathematically, the one-hot encoded matrix is defined as:

$$M_{i,j}^{(k)} = \begin{cases} 1, & \text{if } j = a_i^{(k)} \\ 0, & \text{otherwise} \end{cases} \quad \forall i = 1, \dots, m, j = 1, \dots, L. \quad (53)$$

This gives the final action matrix $\rho(t) = M_{i,j}^{(k)} \in \mathbb{R}^{m \times L}$. For each of the top k actions, a one-hot matrix $\mathbf{M}^{(k)} \in \mathbb{R}^{m \times L}$ is constructed as:

$$\mathbf{M}^{(i)} = \begin{bmatrix} \mathbf{M}_1^{(i)} \\ \mathbf{M}_2^{(i)} \\ \vdots \\ \mathbf{M}_L^{(i)} \end{bmatrix}, \quad i \in \{1, 2, \dots, k\}.$$

The function returns a set of k encoded matrices:

$$\{\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \dots, \mathbf{M}^{(k)}\},$$

where each matrix $\mathbf{M}^{(i)}$ represents the i -th top action for all rows.

4.3.2 Model-based Critic Module

The critic module in this framework is responsible for evaluating the actions $\rho(t)$ taken by the actor module, providing an estimate of the expected rewards based on the current state and the actor's chosen action. This feedback helps the actor improve its decision-making over time by adjusting its policy based on the evaluation provided by the critic. Unlike traditional actor-critic models, where the critic relies on a model-free deep neural network (DNN) to evaluate actions without explicit knowledge of the underlying system dynamics, this framework utilizes a model-based approach. In conventional setups, the critic network is trained solely on experience data and approximates the value function by learning from past actions and rewards. While flexible and applicable to a wide range of problems, this model-free approach may require extensive training and can suffer from instability, especially when dealing with complex resource allocation problems. In contrast, this

framework leverages model-based information to improve the critic's evaluation. It utilizes known system models to solve the optimal resource allocation problem analytically. By solving the optimal resource allocation problem analytically, the framework achieves better performance and faster convergence compared to traditional model-free critics. This approach reduces the uncertainty and variance that can arise from approximation errors in a purely model-free setup, leading to more efficient resource utilization and improved decision-making by the actor.

To solve the optimization problem in (48), given the HDT update allocation parameter $\rho(t)$ which satisfy the constraint C_1 and C_2 , we can rewrite the optimization problem **P1** as

$$\begin{aligned} \mathbf{P2} \quad & \max_{\mathbf{D}(t)} V[\psi(t)] + \sum_{l \in L} [Q_l(t)(\mu_l(t))] - \sum_{l \in L} [Z_l(t)(P_l(t))] \\ & \text{s.t.} \quad C_4 - C_8. \end{aligned} \quad (54)$$

From (31), in each iteration, each encoder can process n HDT updates from their respective queue provided $Q_l(t) \geq n$; hence, for each iteration we have:

$$\begin{aligned} \mathbf{P3} \quad & \sum_{i \in n} \max_{\mathbf{D}(t)} \sum_{l \in L} \left[V \left(\Phi \xi_i^l(t) + \frac{\phi}{\ell_i^l(t)} \right) + Q_l(t)\mu_l(t) - Z_l(t)P_l(t) \right]. \\ & \text{s.t.} \quad C_4 - C_9. \end{aligned} \quad (55)$$

We can therefore optimize resources for each update i by decomposing the problem into an encoding optimization problem and a transmission optimization problem.

Encoding Problem Optimization

Given the allocation action and assume an optimal transmission latency $\bar{\ell}_i^{Tl}(t)$, **P3** can be rewritten as

$$\mathbf{P4} \quad \max_{f_l(t)} \sum_{l \in L} \left[\left(\frac{V\phi}{N_i^l(t)} + Q_l(t) \right) \frac{f_l(t)}{\alpha k_l} - Z_l(t) \kappa(f_l(t))^3 \right] \Big|_i \quad (56a)$$

$$\text{s.t.} \quad \sum_{l \in L} f_l \leq f_{\text{cpu}} \quad (56b)$$

$$\frac{N_i^l(t) \alpha k_l}{f_l(t)} \leq \min \left(\frac{\tau}{n}, \ell_{\text{max}}^i(t) - \ell_i^{Ql}(t) \right) - \bar{\ell}_i^{Tl}(t). \quad (56c)$$

Since $f_l(t) \geq 0$, the optimization problem is concave in the feasible region. We use the Karush-Kuhn-Tucker (KKT) condition to solve it. The Lagrangian function of the **P4** is

$$\begin{aligned} \mathcal{L}(f_l, \lambda, \mu) = & \sum_{l \in L} \left[\left(\frac{V\phi}{N_i^l(t)} + Q_l(t) \right) \frac{f_l}{\alpha k_l} - Z_l(t) \kappa(f_l)^3 \right] \\ & + \lambda \left(f_{\text{cpu}} - \sum_{l \in L} f_l \right) \\ & + \sum_{l \in L} \mu_l \left(\min \left(\frac{\tau}{n}, \ell_{\text{max}}^i(t) - \ell_i^{Ql}(t) \right) - \bar{\ell}_i^{Tl}(t) - \frac{N_i^l(t) \alpha k_l}{f_l} \right), \end{aligned} \quad (57)$$

where λ is the Lagrange multiplier for the total frequency constraint and μ_l is the Lagrange multiplier for the latency constraint of each $l \in L$. The multiplier satisfies the constraints

$$\lambda \geq 0, \quad \mu_l \geq 0 \quad \forall l \in L. \quad (58)$$

Given that (56b) is satisfied, $f_l(t)$ is bounded above by f_{cpu} , and bounded below by

$$f_l^{\text{min}}(t) = \frac{N_i^l(t) \alpha k_l}{\min \left(\frac{\tau}{n}, \ell_{\text{max}}^i(t) - \ell_i^{Ql}(t) \right) - \bar{\ell}_i^{Tl}(t)}. \quad (59)$$

We can obtain $f_l^*(t)$ by finding the root of the stationary point of the Lagrangian function with respect to $f_l(t)$, λ , and μ_l .

Transmission Problem Optimization

The transmission optimization problem, given the update allocation action and the optimized frequency allocation $f_l^*(t)$, can be defined as:

$$\mathbf{P5} \quad \underset{P_i^{Tl}(t), B_l(t)}{max} \quad \sum_{l \in L} \left[V \left(\Phi \xi_i^l(t) + \frac{\phi}{\ell_i^{Tl}(t)} \right) + Q_l(t) \mu_i^{Tl}(t) - Z_l(t) P_i^{Tl}(t) \right] \Big|_i \quad (60a)$$

$$\text{s.t.} \quad \sum_{l \in L} B_l \leq B_{\max} \quad (60b)$$

$$\ell_i^{Tl}(t) \leq \min \left(\frac{\tau}{n}, \ell_{\max}^i(t) - \ell_i^{Ql}(t) \right) - \bar{\ell}_i^{El}(t) \quad (60c)$$

$$P_i^{Tl}(t) \leq P_{\max}^{Tl} \quad (60d)$$

$$\xi_i^l(t) \geq \xi_{\min}^i \quad (60e)$$

The problem **P5** is inherently non-convex, largely due to the nature of semantic communication. In this context, the semantic accuracy, denoted as $\xi_i^l(t)$, is dependent on the machine learning algorithm responsible for interpreting and conveying the meaning of the transmitted data. However, since directly computing the semantic accuracy for each configuration can be complex, we utilize an empirical approximation via the semantic accuracy lookup table provided in [36].

Given the constraints on both accuracy and latency, solving **P5** in its original form presents a challenge. To address this, we first relax the accuracy and latency constraints by incorporating them into a penalty function. This converts the original constrained problem into a more tractable form where deviations from the desired accuracy and latency are penalized, thus softening the hard constraints.

To approximate an optimal solution, we apply differential evolution (DE), a meta-heuristic optimization algorithm well-suited for solving non-convex problems. DE operates by evolving a population of candidate solutions through the mechanisms of mutation, crossover, and selection. Each

candidate solution is iteratively refined based on its performance, with the goal of minimizing the objective function while accounting for the penalty terms associated with accuracy and latency.

Differential evolution is particularly effective in this context because it is capable of exploring large, complex solution spaces without being restricted by convexity assumptions. It uses a population-based approach, allowing it to maintain diversity in the search space and avoid being trapped in local optima. Over successive generations, DE converges towards a solution that satisfies the relaxed constraints while approximating the optimal trade-off between accuracy, latency, and power consumption. This combined approach—relaxation of constraints via a penalty function and optimization via differential evolution—offers a robust and efficient strategy for solving **P5**, even in the face of the problem’s non-convexities and the dependencies on machine learning models for semantic accuracy.

4.3.3 DNN Policy Update Module

This module is responsible for training the Deep Neural Network (DNN) model to update the policy for data allocation decisions. It operates by randomly selecting samples from a replay memory, which stores recent data pairs, $(g(t), \rho(t))$, where $g(t)$ represents the observed state and $\rho(t)$ is the corresponding data allocation decision. The replay memory has a fixed size and only retains the most recent training samples, starting initially as an empty buffer and filling up over time as new data is gathered.

The DNN model is initialized randomly and used to generate initial data allocation decisions. The policy is updated at regular intervals to continuously improve the quality of these decisions. To avoid overfitting, the DNN is retrained every δ_T time units, which allows the model to adapt without becoming too dependent on recent data samples.

$$\mathcal{F}(\theta(t)) = -\frac{1}{|S(t)|} \sum_{\tau \in S(t)} \left[\rho^*(\tau)^\top \log f_{\theta(t)}(g(\tau)) + (1 - \rho^*(\tau))^\top \log (1 - f_{\theta(t)}(g(\tau),)) \right]. \quad (61)$$

At each retraining point, when $(t \bmod \delta_T) = 0$, a random batch of data samples is drawn from the

set $\{(g(\tau), \rho(\tau)) \mid \tau \in S(t)\}$, where $S(t)$ denotes the available time indices in the replay memory. The Adam optimizer is then used to update the DNN's parameters by minimizing the average cross-entropy loss over the selected samples. This ensures that the DNN continues to refine its ability to make accurate data allocation decisions over time. We summarize the whole procedure of the proposed DRL-Based HDT Synchronization Algorithm in Algorithm 2.

Algorithm 2 DRL-Based HDT Synchronization Algorithm

- 1: **Input:** Parameters V, T , training interval δ_T , $k(t)$ update interval δ_k ;
 - 2: **Output:** Control actions $\{\mathbf{A}(t)\}_{t=1}^T$;
 - 3: Initialize the DNN with random parameters and empty replay memory, $k \leftarrow L$;
 - 4: Empty initial data queue $Q_l(1) = 0$ and power queue $Z_l(1) = 0$, for $i = 1, \dots, L$;
 - 5: **for** $t = 1, 2, \dots, T$ **do**
 - 6: Observe the input $g(t) = \{h(t), \mathbf{Q}(t), \mathbf{Z}(t)\lambda(t)\}$ and update $k(t)$ if $\text{mod}(t, \delta_k) = 0$;
 - 7: Generate a relaxed data allocation action $O(t) = f_{\theta(t)}(g(t))$ with the DNN;
 - 8: convert the relax action into probability $Pr_{i,j}(t) \in m, j \in L$ using softmax
 - 9: Select top $k(t)$ actions using order preserving method and quantize $P_{i,j}(t)$ into $k(t)$ one-hot binary actions $\rho(t)$;
 - 10: **for** $i = 1, 2, \dots, k$ **do**
 - 11: Use the i th action to execute the synchronization mini slots, and optimize the resource allocation $\mathbf{D}(t)$
 - 12: **end for**
 - 13: Select the best action $\mathbf{A}^*(t)$ that gives the maximum reward
 - 14: Update the replay memory by adding $(g(t), \rho(t))$;
 - 15: **if** $\text{mod}(t, \delta_T) = 0$ **then**
 - 16: Uniformly sample a batch of data set $\{(g(\tau), \rho(\tau)) \mid \tau \in S(t)\}$ from the memory;
 - 17: Train the DNN with $\{(g(\tau), \rho(\tau)) \mid \tau \in S(t)\}$ and update $\theta(t)$ using the Adam optimizer;
 - 18: **end if**
 - 19: update the queue $Q_l(t)$ and $Z_l(t)$ state
 - 20: $t \leftarrow t + 1$;
 - 21: new data arrival $\lambda(t)$ and channel state $h(t)$;
 - 22: **end for**
-

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This work introduces an innovative multi-encoder semantic communication model for HDT synchronization, which takes advantage of the diversity in HDT update dynamics. By configuring multiple encoders with varying capacities and configurations, the model allows for dynamic allocation of PT updates to the most appropriate encoder. This decision is based on the available resources and the specific requirements of each update, enabling a more flexible and efficient system design.

The system is formulated as a MINLP optimization problem, aiming to maximize the overall system performance. The performance is defined as a linear combination of latency and accuracy metrics, which are essential for ensuring the successful transmission and interpretation of HDT data. To solve this problem, an optimal solution is derived using a Genetic Algorithm (GA), a technique known for its ability to handle complex, nonlinear optimization tasks with mixed-integer variables.

The simulation results presented demonstrate that the proposed multi-encoder semantic system significantly outperforms existing semantic communication models. Specifically, it excels in terms of power management, ensuring that system resources are utilized efficiently. Additionally, the proposed model achieves optimal latency and accuracy, providing a robust solution for HDT synchronization in resource-constrained environments.

We extend the short-term optimization problem into a long-term framework to ensure both stability and optimal decision-making that considers future objectives. Using Lyapunov optimization,

we break the problem into manageable deterministic subproblems for each time step while maintaining long-term constraints. To solve this, we employ a hybrid deep reinforcement learning approach, combining a model-free actor for decision-making and a model-based critic for evaluating actions. This approach optimizes resource allocation over time, balancing immediate and future performance goals effectively.

5.2 Future Work

The future work highlighted in this thesis is divided into two sections and presented below.

5.2.1 Developing and Simulating the framework "Lyapunov-Assisted DRL-based Multi-Encoder Semantic Communication for HDT Synchronization" as outlined in Chapter 4

This simulation will provide insights into how effectively the Lyapunov-assisted reinforcement learning mechanism can optimize HDT synchronization, given the dynamic demands and complexities of real-time environments. The simulation should focus on verifying the framework's performance across different metrics, including resource utilization, latency, and system stability. By testing multiple encoder configurations and observing how the Lyapunov-based decision-making manages short-term resource demands alongside long-term stability objectives, the simulation will highlight the framework's efficiency in adapting to varying workloads and ensuring reliable HDT update synchronization. Furthermore, the model-free actor and model-based critic components within the DRL setup can be tested to evaluate how well they cooperate in achieving an optimal balance between immediate and long-term goals under constraints.

5.2.2 Preemptive Queue Protocol

It is also important to explore the integration of a preemptive queue protocol to further enhance the efficiency of the framework. In the current setup, updates are processed in the order of their arrival, which may not always align with the urgency or importance of the updates. By implementing a preemptive queue protocol, we can prioritize HDT updates based on their specific requirements,

regardless of their position in the queue. This would ensure that critical updates with strict deadlines or higher priority are processed first, thereby improving the system's responsiveness and its ability to meet real-time constraints.

In addition to considering the preemptive queue protocol, it is essential to also explore the implementation of a breakout mechanism to prevent updates from remaining in the queue indefinitely. Without such a mechanism, there is a risk of congestion in the queue, leading to delays beyond acceptable limits. The breakout mechanism would impose a time constraint, ensuring that updates are not left in the queue for too long without being processed. This would not only prevent delays from accumulating but also ensure a continuous and smooth flow of updates, reducing the risk of bottlenecks in the system.

Together, the incorporation of a preemptive queue protocol and a breakout mechanism could significantly improve the framework's adaptability to varying workloads and real-time demands, enhancing its overall efficiency, stability, and timeliness in HDT synchronization tasks. These enhancements would better enable the system to handle resource constraints and dynamic environments in future implementations.

Bibliography

- [1] J. Chen, C. Yi, S. D. Okegbile, J. Cai, and X. S. Shen, “Networking architecture and key supporting technologies for human digital twin in personalized healthcare: a comprehensive survey,” *IEEE Communications Surveys & Tutorials*, 2023.
- [2] S. D. Okegbile, J. Cai, D. Niyato, and C. Yi, “Human digital twin for personalized healthcare: Vision, architecture and future directions,” *IEEE network*, vol. 37, no. 2, pp. 262–269, 2022.
- [3] J. Chen, C. Yi, H. Du, D. Niyato, J. Kang, J. Cai, and X. Shen, “A revolution of personalized healthcare: Enabling human digital twin with mobile aigc,” *IEEE Network*, 2024.
- [4] S. D. Okegbile, J. Cai, H. Zheng, J. Chen, and C. Yi, “Differentially private federated multi-task learning framework for enhancing human-to-virtual connectivity in human digital twin,” *IEEE Journal on Selected Areas in Communications*, 2023.
- [5] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, “Digital twin for 5g and beyond,” *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.
- [6] C. K. Thomas, W. Saad, and Y. Xiao, “Causal semantic communication for digital twins: A generalizable imitation learning approach,” *IEEE Journal on Selected Areas in Information Theory*, 2023.
- [7] J. Moyne, Y. Qamsane, E. C. Balta, I. Kovalenko, J. Faris, K. Barton, and D. M. Tilbury, “A requirements driven digital twin framework: Specification and opportunities,” *Ieee Access*, vol. 8, pp. 107781–107801, 2020.

- [8] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2020.
- [9] S. Okegbile, H. Gao, O. Talabi, J. Cai, C. Yi, D. Niyato, and X. S. Shen, "Fles: A federated learning-enhanced semantic communication framework for mobile aigc-driven human digital twins," *Authorea Preprints*, 2024.
- [10] J. Tang, J. Nie, J. Bai, J. Xu, S. Li, Y. Zhang, and Y. Yuan, "Uav-assisted digital twin synchronization with tiny machine learning-based semantic communications," *IEEE Internet of Things Journal*, 2024.
- [11] D. Gelernter, *Mirror worlds: Or the day software puts the universe in a shoebox... How it will happen and what it will mean*. Oxford University Press, 1993.
- [12] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary perspectives on complex systems: New findings and approaches*, pp. 85–113, 2017.
- [13] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital twin: Origin to future," *Applied System Innovation*, vol. 4, no. 2, p. 36, 2021.
- [14] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.
- [15] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [16] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," *Computers in industry*, vol. 113, p. 103130, 2019.
- [17] L. Bao, Q. Wang, and Y. Jiang, "Review of digital twin for intelligent transportation system," in *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*, pp. 309–315, IEEE, 2021.

- [18] M. S. Irfan, S. Dasgupta, and M. Rahman, "Towards transportation digital twin systems for traffic safety and mobility: A review," *IEEE Internet of Things Journal*, 2024.
- [19] T. Ruohomäki, E. Airaksinen, P. Huuska, O. Kesäniemi, M. Martikka, and J. Suomisto, "Smart city platform enabling digital twin," in *2018 International Conference on Intelligent Systems (IS)*, pp. 155–161, IEEE, 2018.
- [20] M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi, "A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future," *IEEE Access*, vol. 11, pp. 17471–17484, 2023.
- [21] M. E. Miller and E. Spatz, "A unified view of a human digital twin," *Human-Intelligent Systems Integration*, vol. 4, no. 1, pp. 23–33, 2022.
- [22] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, and S. Valtolina, "Human digital twin for fitness management," *Ieee Access*, vol. 8, pp. 26637–26664, 2020.
- [23] Q. Wang, W. Jiao, P. Wang, and Y. Zhang, "Digital twin for human-robot interactive welding and welder behavior analysis," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 334–343, 2020.
- [24] I. Graessler and A. Pöhler, "Integration of a digital twin as human representation in a scheduling procedure of a cyber-physical production system," in *2017 IEEE international conference on industrial engineering and engineering management (IEEM)*, pp. 289–293, IEEE, 2017.
- [25] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE access*, vol. 7, pp. 49088–49101, 2019.
- [26] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *Ieee Access*, vol. 7, pp. 20325–20336, 2019.
- [27] J. Zhang, L. Li, G. Lin, D. Fang, Y. Tai, and J. Huang, "Cyber resilience in healthcare digital twin on lung cancer," *IEEE access*, vol. 8, pp. 201900–201913, 2020.

- [28] Z. Hu, S. Lou, Y. Xing, X. Wang, D. Cao, and C. Lv, "Review and perspectives on driver digital twin and its enabling technologies for intelligent vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 417–440, 2022.
- [29] R. Martinez-Velazquez, R. Gamez, and A. El Saddik, "Cardio twin: A digital twin of the human heart running on the edge," in *2019 IEEE international symposium on medical measurements and applications (MeMeA)*, pp. 1–6, IEEE, 2019.
- [30] H. Elayan, M. Aloqaily, and M. Guizani, "Digital twin for intelligent context-aware iot health-care systems," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16749–16757, 2021.
- [31] K. Amara, O. Kerdjadj, and N. Ramzan, "Emotion recognition for affective human digital twin by means of virtual reality enabling technologies," *IEEE Access*, vol. 11, pp. 74216–74227, 2023.
- [32] S. D. Okegbile and J. Cai, "Edge-assisted human-to-virtual twin connectivity scheme for human digital twin frameworks," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, pp. 1–6, IEEE, 2022.
- [33] X. Li, B. He, Z. Wang, Y. Zhou, G. Li, and R. Jiang, "Semantic-enhanced digital twin system for robot–environment interaction monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.
- [34] B. Du, H. Du, H. Liu, D. Niyato, P. Xin, J. Yu, M. Qi, and Y. Tang, "Yolo-based semantic communication with generative ai-aided resource allocation for digital twins construction," *IEEE Internet of Things Journal*, 2023.
- [35] Z. Ji and Z. Qin, "Computational offloading in semantic-aware cloud-edge-end collaborative networks," *arXiv preprint arXiv:2402.18183*, 2024.
- [36] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, "Resource allocation for text semantic communications," *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1394–1398, 2022.

- [37] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Ye Li, “Qoe-aware resource allocation for semantic communication networks,” in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 3272–3277, 2022.
- [38] A. Yekanlou, A. I. Salameh, and J. Cai, “Buffer-state aware task offloading in edge networks with task splitting for iov,” in *2023 Biennial Symposium on Communications (BSC)*, pp. 13–18, IEEE, 2023.
- [39] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, “Deep reinforcement learning for stochastic computation offloading in digital twin networks,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4968–4977, 2020.
- [40] S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, “Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7519–7537, 2021.