University of Glasgow

Meng, Zhen (2023) *Task-oriented cross-system design for Metaverse in 6G era.* PhD thesis.

https://theses.gla.ac.uk/83985/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

Task-Oriented Cross-System Design for Metaverse in 6G Era

Zhen Meng

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

August 2023

Dedicated to my family, teachers, and friends.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgments. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables, and equations, and has fewer than 150 figures.

Abstract

As an emerging concept, the Metaverse has the potential to revolutionize social interaction in the post-pandemic era by establishing a digital world for online education, remote healthcare, immersive business, intelligent transportation, and advanced manufacturing. The goal is ambitious, yet the methodologies and technologies to achieve the full vision of the Metaverse remain unclear. In this thesis, we first introduce the three pillars of infrastructure that lay the foundation of the Metaverse, i.e., Human-Computer Interfaces (HCIs), sensing and communication systems, and network architectures. Then, we depict the roadmap towards the Metaverse that consists of four stages with different applications. As one of the essential building blocks for the Metaverse, we also review the state-of-the-art Computer Vision for the Metaverse as well as the future scope. To support diverse applications in the Metaverse, we put forward a novel design methodology: task-oriented cross-system design, and further review the potential solutions and future challenges.

Specifically, we establish a task-oriented cross-system design for a simple case, where sampling, communications, and prediction modules are jointly optimized for the synchronization of the real-world devices and digital model in the Metaverse. We use domain knowledge to design a deep reinforcement learning (DRL) algorithm to minimize the communication load subject to an average tracking error constraint. We validate our framework on a prototype composed of a real-world robotic arm and its digital model. The results show that our framework achieves a better trade-off between the average tracking error and the average communication load compared to a communication system without sampling and prediction. For example, the average communication load can be reduced to 87% when the average track error constraint is 0.002°. In addition, our policy outperforms the benchmark with the static sampling rate and prediction horizon optimized by exhaustive search, in terms of the tail probability of the tracking error. Furthermore, with the assistance of expert knowledge, the proposed algorithm achieves a better convergence time, stability, communication load, and average tracking error.

Furthermore, we establish a task-oriented cross-system design framework for a general case, where the goal is to minimize the required packet rate for timely and accurate modeling of a real-world robotic arm in the Metaverse. Specifically, different modules including sensing, communications, prediction, control, and rendering are considered. To optimize a scheduling policy and prediction horizons, we design a Constraint Proximal Policy Optimization (CPPO) algorithm by integrating domain knowledge from relevant systems into the advanced reinforcement learning algorithm, Proximal Policy Optimization (PPO). Specifically, the Jacobian matrix for analyzing the motion of the robotic arm is included in the state of the CPPO algorithm, and the Conditional Value-at-Risk (CVaR) of the state-value function characterizing the long-term modeling error is adopted in the constraint. Besides, the policy is represented by a two-branch neural network determining the scheduling policy and the prediction horizons, respectively. To evaluate our algorithm, we build a prototype including a real-world robotic arm and its digital model in the Metaverse. The experimental results indicate that domain knowledge helps to reduce

the convergence time and the required packet rate by up to 50%, and the cross-system design framework outperforms a baseline framework in terms of the required packet rate and the tail distribution of the modeling error.

Publications

Throughout this PhD career, three publications have been accepted or submitted of which the following are related to this thesis:

- Z. Meng, C. She, G. Zhao, M. A. Imran, M. Dohler, Y. Li, and B. Vucetic, "Task-Oriented Metaverse Design in the 6G Era," to be appeared in *IEEE Wireless Communications, 2023*
- Z. Meng, C. She, G. Zhao and D. De Martini, "Sampling, communications, and Prediction Co-Design for Synchronizing the Real-World Device and Digital Model in Metaverse," in *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 288-300, Jan. 2023. (This paper was selected by the Editor-in-Chief as the featured paper, more details can be found in IEEE JSAC website:^_ https : //www.comsoc.org/publications/blogs/selected – ideas – communications/synchronizing – physical – devices – metaverse)
- Z. Meng, K Chen, Y, Diao, C. She, G. Zhao, M. A. Imran, and B. Vucetic, "Task-Oriented Cross-System Design for Timely and Accurate Modeling in the Metaverse," second-round review in *IEEE Journal on Selected Areas in Communications*, 2023.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

5G Fifth Generation

6G Sixth Generation

ADE Average Displacement Error

AGV Autonomous Guided Vehicle

AI Artificial Intelligence

AoI Age of Information

AP Average Precision

APDO Accelerated Primal-dual Policy Optimization

AR Augmented Reality

ASD-O Absolute Scale Difference - Object

ASD-V Absolute Scale Difference - Viewpoint

C-PPO Constraint Proximal Policy Optimization

CARLA Car Learning to Act

CCDF Complementary Cumulative Distribution Function

CMDP Constrained Markov Decision Processes

CNN Convolutional Neural Network

CRPO Constraint-Rectified Policy Optimization

CV Computer Vision

CVaR Conditional Value-at-Risk

D-H Denavit–Hartenberg

LIST OF ABBREVIATIONS

DDPG Deep Deterministic Policy Gradient

DoF degrees of freedom

DRL Deep Reinforcement Learning

DTs Digital twins

E2E End-to-End

eMBB Enhanced Mobile Broadband

FDE Final Displacement Error

FID Fréchet Inception Distance

FL Federal Learning

GAE Generalized Advantage Estimate

HCI Human-Computer Interface

HDMI High-Definition Multimedia Interface

IIoT Industrial Internet of Things

IoT Internet of Things

IoU Intersection over Union

ISAC Integral Sensing and communications

JND Just Noticeable Difference

JSCC Joint Source and Channel Coding

KC-TD3 Knowledge-assisted Constrained Twin-Delayed Deep Deterministic

KPI Key Performance Indicator

LPIPS Learned Perceptual Image Patch Similarity

LSTF Long Sequence Time-series Forecast

LSTM Long Short-Term Memory

MA Moving Average Method

MAE Mean Absolute Error

mAP mean Average Precision

mAPH mean Average Precision with Hierarchy

MEC Multi-access Edge Computing

MFCC Mel Frequency Cepstral Coefficient

MI Mutual Information

mIoU mean IoU

MLP Multi-Layer Perception

mMTC Massive Machine Type communications

MOTA Multiple Object Tracking Accuracy

MOTA Multiple Object Tracking Accuracy

MPJPE Mean Per Joint Position Error

MR Mixed Reality

MS-SSIM Multi-Scale Structural Similarity Index

MSE Mean Squared Error

MSE Mean Squared Error

MTP Motion-To-Photon

NFT Non-Fungible Token

O-RAN Open-Radio Access Networks

OA Overall Accuracy

OFDM Orthogonal Frequency-Division Multiplexing

PCK Percentage of Correct Keypoint

PCK Percentage of Correct Keypoints

PCKh Percentage of Correct Keypoints with a normalized head size

PD Proportional–Derivative

PPO Proximal Policy Optimization

LIST OF ABBREVIATIONS

PSNR Peak Signal-to-Noise Ratio

PSNR Peak Signal-to-Noise Ratio

QoS Quality of Service

RAN Radio Access Network

RMP Riemannian Motion Policy

ROS Robot Operating System

SLAM Simultaneous Localization And Mapping

SSIM Structural Similarity Index Measure

TRPO Trust Region Policy Optimization

UR3e Universal Robot of the UR3e e-Series

URLLC Ultra-Reliable Low-Latency communications

VaR Value-at-Risk

VOR Vestibulo-Ocular Reflex

VPG Vanilla Policy Gradient

VR Virtual Reality

WNCS Wireless Networked Control System

XR Extended Reality

Chapter 1

Introduction

1.1 Why the Metaverse?

The Metaverse, fueled by the information "Big Bang," stands out as a potent force driving digital transformation and is poised to succeed the current network paradigm. Its boundless potential extends to a complete restructuring of both physical and digital realms, with a particular emphasis on shaping the architecture of future communication systems [1]. The concept of the Metaverse was initially introduced in Neil's book, *Snow Crash* [2], coinciding with the development of virtual physical fusion technology. Specifically, it is recognized as a digital world that will revolutionize the interactions among humans, machines, and environments by providing a shared, unified, perpetual, and inter-operable realm for participants from all over the world [3]. The digital world could be a purely virtual space or a digital mirror of the physical world that has the ability to reprogram the physical world in real time. It lays the foundation for the evolution of different vertical industries including education, entertainment, healthcare, manufacturing, transportation, and immersive business. In fact, the contemporary landscape of communication technology is witnessing a profound synchronic convergence with the digital transformation process. On the one hand, the medium of transmission is transformed from simple formats (e.g., text, sound, and picture) with limited options of user engagement (like, retweet) to a variety of forms and storage of rich information (e.g., HD video, point cloud data, Non-Fungible Token (NFT), etc.) and diverse operations (haptic control, teleportation). Meanwhile, digital transformation drives the paradigm shift from the "digital island" represented by the typical standalone XR to a universal and ubiquitous duality of things in every aspect of life [4].

The Metaverse aims to revolutionize social interaction by providing an immersive environment for online activities. It can provide new opportunities for business, such as virtual real estate, in-world advertising, and virtual product sales. It is also believed that Metaverse can promote remote education by providing a personalized and immersive learning experience. Furthermore, the Metaverse can enhance healthcare as it will offer new ways for healthcare services and support virtual consultations, remote monitoring, and telemedicine. Finally, the Metaverse can be applied

in factory automation by establishing a digital factory to monitor and reprogram the real-world factory [5].

- **Enhanced social interaction**: The Metaverse could provide a more immersive and engaging way for people to connect and socialize online. With the Metaverse, people could interact with each other as if they were in the same physical space, even if they are geographically dispersed [6].
- **Developing Business Opportunities**: The Metaverse could create new opportunities for businesses, such as virtual real estate, in-world advertising, and virtual product sales. These new opportunities could potentially create new revenue streams for companies.
- **Promoting Education**: The Metaverse could offer new ways to provide education and training. It could provide immersive, interactive, and personalized learning experiences, allowing students to learn at their own pace and in their own way. It could also facilitate collaboration and communications between students and teachers, regardless of their physical location [7].
- **Enhance Healthcare**: The Metaverse could offer new ways to deliver healthcare services and support. It could enable virtual consultations, remote monitoring, and telemedicine, allowing healthcare providers to reach more patients and provide more personalized care.

1.2 Task-Oriented Cross-System Design for the Metaverse

This ambitious vision brings significant challenges to the development of next-generation communication networks. It is natural to raise the following questions:

Q1: Is the available infrastructure sufficient for the Metaverse?

To support an application in the Metaverse, the system needs to execute a sequence of interdependent tasks. A task is an activity that needs to be completed within a period of time or by a deadline, such as pose and eye tracking, positioning, haptic control and feedback, and semantic segmentation [4]. The state-of-the-art infrastructure cannot meet the requirements of diverse emerging applications and tasks in the Metaverse. Specifically, existing input/output systems, such as the touch screen, keyboard, and mouse, are inconvenient in supporting new tasks. Thus, the new HCI, including VR, AR, Tactile Internet, and brain-computer interface, will lay the foundation for the Metaverse. Sensing and communication technologies play critical roles in providing timely feedback and seamless connections in the Metaverse with a real-world counterpart. To reduce infrastructure costs, a promising approach is to exploit widely deployed mobile networks for both sensing and communications. Furthermore, the Sixth Generation (6G) networks will bridge new HCI and sensing & communication systems. Due to the long propagation delay, executing all tasks on a global server cannot meet the latency requirements of tasks. A new multi-tier network

architecture that can efficiently coordinate computing, communications, and storage resources on the end-user devices, edge/local servers, and global servers is essential to support interdependent tasks in the Metaverse [8]. In summary, HCI, sensing, communication technologies, and network architectures will serve as the three pillars of the Metaverse. Even with the above infrastructure, supporting emerging applications in the Metaverse is not straightforward.

Q2: How to guarantee the KPIs of diverse applications/tasks in the Metaverse?

The highly integrated and multifaceted demands of applications in the Metaverse impose stringent requirements on KPIs that are much more diverse than KPIs defined in the three typical scenarios in the Fifth Generation (5G) mobile communication standard, i.e., Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency communications (URLLC), and Massive Machine Type communications (mMTC) [9]. Furthermore, considering that one application consists of multiple tasks, to meet the specific requirements of the application in the Metaverse, we should analyze the KPIs at the task level, referred to as task-oriented KPIs. For example, to generate haptic feedback, the system should meet the Just Noticeable Difference (JND) constraint, which is the minimum difference between two force signals that is noticeable to users. The network functions and communications KPIs in 5G networks are task agnostic and hence cannot guarantee task-level KPIs.

The existing communication network design approach divides the whole system into multiple sub-modules for separate optimization and cannot break the barriers among the sub-modules. As a result, it is difficult to provide E2E performance guarantees. To support the Metaverse in 6G mobile networks, we should revisit the following questions:

Q3: What are the issues with the existing design approaches? Do we need new design methodologies in 6G?

To improve E2E performance, the design of the system between systems has been investigated in the existing literature [10]–[12]. To guarantee control performance with stochastic wireless channels and limited communication resources in mission-critical applications, a predictive control and communication co-design system was introduced in [13], where the scheduling policy and transmission power are jointly optimized. To achieve substantial gains in spectral, energy, hardware, and cost efficiency with mMTC, Integral Sensing and communications (ISAC) was developed in [11] to support sensing and communications simultaneously. Considering that end-user devices have limited computing, communications, and storage resources, a cloud-edge-end computing framework-driven solution was introduced in [12]. However, the cross-system design problems are in general non-convex or NP-hard, and novel design methodologies for real-time implementation are in urgent need.

Q4: Why Do We Need Task-Oriented Design for the Metaverse?

To realize the grand vision of the Metaverse, we can continue the never-ending race towards wider bandwidth and higher frequency bands, optimizing existing communication architectures. However, the wealth of personalized content in Metaverse that served as the catalyst calls for a

more efficient communication resource utilization architecture that is capable of leveraging the diversity of user demands. The richness of the Metaverse in terms of applications and transmission media motivates us to rethink the three questions proposed in Shannon's information theory, especially the second and third questions, i.e., the task-oriented semantics of transmission and its post-transmission effects [14]. The Metaverse's interconnectedness, shared data storage, and Artificial Intelligence (AI)-empowering technology provide us more opportunities to think rather than reconstruct the underlying message, but to enable the receiver to make the right inference or to take the right action at the right time and context. In this regard, designing the communication system from a task-oriented perspective is logical and natural. The communication paradigm can shift from guaranteeing the correct reception of each transmitted bit, irrespective of the meaning conveyed by the transmitted bits, to focusing on transmitting bits that are key to accomplishing the goals or tasks by analyzing the semantic context or relevant information [15]. However, how to apply task-oriented design to the Metaverse is still nascent with many fundamental problems yet to be investigated in sufficient depth.

Q5: Why Do We Need Cross-System Design for the Metaverse?

The communication system design alone cannot fulfill all the KPIs of the diverse applications and tasks in the Metaverse. This is because the highly integrated and multifaceted demands of Metaverse applications impose stringent requirements on KPIs that are much more diverse than the KPIs defined in the three typical scenarios in the 5G mobile communication standard, i.e., eMBB, URLLC, and mMTC. Furthermore, as shown in Fig. 1.1, the three pillar technologies of the Metaverse, i.e., HCI, sensing and communications, and network architecture are deeply coupled, where the intermediate signaling processes including information generation, processing, communications, and utilization (such as reconstruction, control, rendering, actuation). Existing HCI, sensing, communications, and computing systems are developed separately [4]. This design approach leads to sub-optimal solutions, brings extra communication overhead for coordinating multiple tasks, and cannot meet task-oriented KPIs [16]. Cross-system problems are in general very complicated and may not have well-established models. As a result, most of the existing analytical tools and optimization algorithms are not applicable. To implement the cross-system design methodology for Metaverse in 6G, the clear cross-system design framework, optimization strategy, and fundamental trade-offs between system-level efficiency and performance need to be investigated.

1.3 Pillars of the Metaverse

1.3.1 Human-Computer Interface

Traditional HCI relies on keyboards, mice, touch screens, etc., and only generates two-dimensional video or audio signals. However, they are inconvenient to support diverse tasks in Metaverse, such

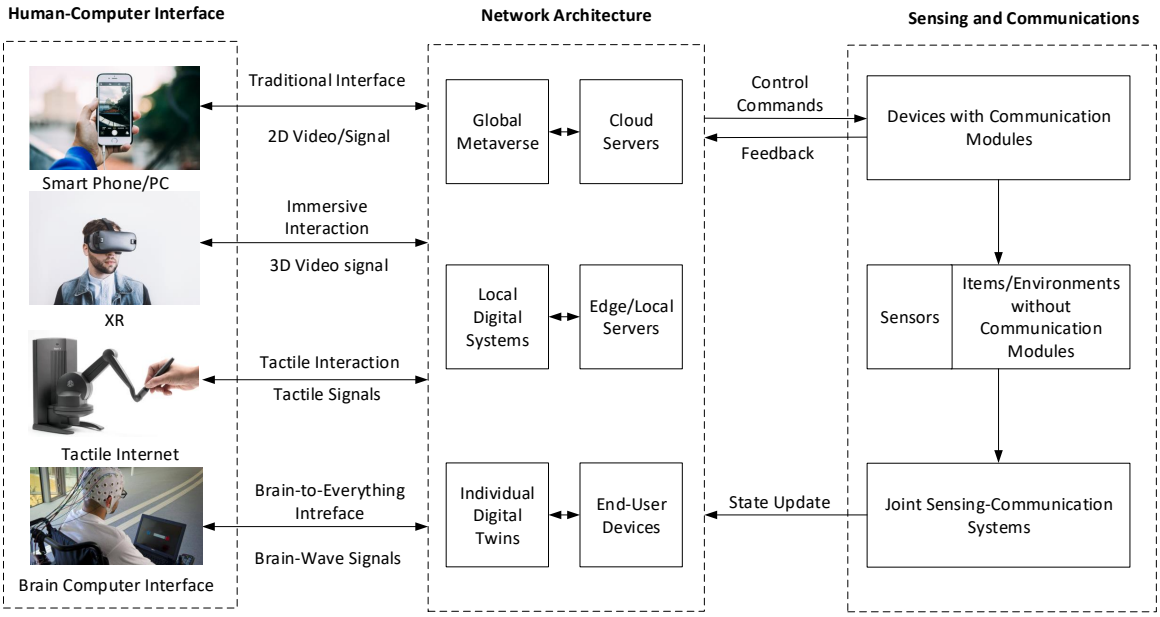


Figure 1.1: Three pillars of the Metaverse, i.e., HCI, sensing and communications, and network architecture.

as pose and eye tracking, positioning, haptic control and feedback, and semantic segmentation. Different from existing input/output systems that are designed to process video and audio signals, future HCI should be carefully designed to support these new tasks in the Metaverse.

- *XR Head-Mounted Devices*: The development of XR devices has greatly improved the user experience by identifying the mobility of the head-mounted device and rendering the three-dimensional (3D) video accordingly. Existing XR systems mainly focus on downlink video streaming. To further enable eye contact and expression reconstruction in the Metaverse, eye tracking, and 3D modeling techniques should be integrated into XR systems. By predicting the moving direction of eyes, the XR system can render and transmit the field-of-view to be requested by users. Thus, we can improve the trade-off between data rate and latency in wireless XR.

- *Tactile Devices*: Tactile devices are essential for supporting haptic feedback in the Metaverse. With a large number of tactile sensors and actuators, it is possible to recognize users' poses and gestures. Once the user hits a virtual item in the Metaverse, the tactile devices generate feedback to users via vibrations and resistance. Most existing tactile devices cannot provide tactile feedback for the entire human body. Several issues remain open in the development of whole-body tactile devices: 1) the battery lifetime of wearable devices is limited; 2) low-complexity graph signal processing that takes the topology of the sensors/actuators is not available; 3) the actuators should be controlled by engines and algorithms to mimic the tactile experience, which remains an open challenge.

- *Brain-Computer Interface*: The brain-computer interface can be used for emotion recognition and reconstruction in the Metaverse. Existing brain-computer interfaces suffer from low classification accuracy and long processing delay. Due to these issues, the brain-computer interface may not be able to work as the stand-alone HCIs in the near future, but it may assist VR devices or tactile devices to improve user experience, as demonstrated in early trials by Meta.

- *Combination of Different Human-Computer Interfaces*: Different HCIs have different data structures, generate responses in different time scales, and may support different tasks in one application. Developing a system that manages multiple HCIs brings unprecedented challenges and is crucial for improving the user experience in the Metaverse. To enable interactions among users with different types of devices, new standards are needed.

1.3.2 Sensing and Communications

Sensing and communication technologies enable timely state updates of real-world devices and environments in the Metaverse. We expect the sensing and communication technologies to provide: (1) Reconstructing the real environment into the virtual world to gain immersive feelings undistinguished between the real and the virtual; (2) Understanding the environment and empowering users with capabilities that they do not have in the real world; (3) Sharing information anywhere, anytime, to gain the ability to communicate in an unobstructed and comprehensive way. Thus, they are critical to the establishment of the digital world. We classify devices and environments into two categories: devices with communication modules and environments without communication modules.

- *Devices with communication modules*: Smart devices equipped with communication modules can update their states to the Metaverse. For example, a real-world robotic arm measures the angles, speeds, forces, and torques of the joints and sends the states to a server for reconstructing the digital robotic arm. As the number of devices increases, the communication resources become the bottleneck of the Metaverse. Improving the trade-off between the communication resource utilization efficiency and the synchronization accuracy/information freshness is a challenging problem.

- *Environments without communication modules*: Some entities in real-world environments do not have communication modules, such as trees, buildings, pedestrians, etc. To collect their states in the digitally twinned Metaverse, we need a large number of external sensors or cameras. For example, Instant-Nerf is a neural rendering model developed by NVIDIA that can render 2D photos into 3D scenes in a few milliseconds [17]. To further understand the environment, semantic segmentation is crucial [18]. Nevertheless, most of the existing segmentation algorithms require a considerable amount of computation resources, and

processing time remains the bottleneck of real-time interactions. Therefore, how to achieve: (1) fast and compact while capturing high-frequency 3d model reconstruction (Sensing), (2) accurate understanding of complex and changing Metaverse scenes (semantic segmentation and object detection), (3) reconstructing models to meet stringent physical and functional constraints is the main challenge.

- *Joint Sensing-communication systems*: Complex environmental perception and inference in the Metaverse and ubiquitous communications make a paradigm that combined perception and communications become one of the possible solutions. The cost of deploying and operating a large number of sensors and cameras could be extremely high. By integrating communications and sensing functionalities into widely deployed mobile networks, it is possible to reduce the cost. Thus, the ISAC system is a practical approach that collects real-world information for the Metaverse [11]. Note that there are tradeoffs between the KPIs of different tasks and the resource utilization efficiency of ISAC systems, but a universal design framework for different tasks is still missing.

1.3.3 Network Architecture

Developing the Metaverse on a global server for all the users and devices around the world would be very challenging due to long communication delay and limited communication throughput. A sequence of interdependent tasks should be executed at the central server, edge server, or end-user device [19]. Thus, the 6G core networks need new network functions for task-level resource management and task offloading.

Multi-tier computing is believed to be a promising architecture that can coordinate interdependent tasks in the Metaverse by exploiting distributed computing, storage, and communication resources in central servers, local servers, and end-user devices [8]. For the diverse applications of the Metaverse and their respective characteristics, multi-tier network architectures can be used to not only maximize user-personalized requirements but also take full advantage of the resources available along its continuum. By distributing communications, computational, and storage capabilities anywhere between the cloud and the user, the potential of a range of computation and latency-intensive applications in the Metaverse will be unlocked [20]. For example, in VR streaming services, two streams of 2D videos are transmitted to a head-mounted device, where they are converted to 3D video. To reduce the weight of head-mounted devices, an alternative approach is to render the 3D video on the local server equipped at the base station or the wireless access point. In this way, we can remove the processing units from the head-mounted devices, and improve the battery lifetime [21].

In addition, the stark contrast between the demanding computing and communication capacity requirements of the Metaverse applications and the trend toward mobility and miniaturization on the client side presents a serious challenge to the current network architecture. (1) It is infeasible

to achieve the global Metaverse by deploying centralized architecture. (2) Diversity applications in the Metaverse vary greatly for computing and communications, which means not all media exchange needs real-time performance. Thus, by executing distributed deep learning algorithms in the multi-tier architecture, the communication overhead for state updates can be reduced significantly. Nevertheless, improving the overall performance with local observations or limited information exchange brings significant challenges in distributed algorithm design and network management.

Meanwhile, multi-tier computing raises new challenges in 6G core networks and Radio Access Networks (RANs).

- *Core Networks*: 5G core networks manage resources and quality-of-service at the application level. The session management function will create a new protocol data unit session when there is a new service request. How to coordinate multiple tasks for one application remains unclear. To address this issue, several promising techniques have been investigated in the existing literature: 1) The authors of [22] developed a semantic-effectiveness plane task-level information processing. 2) In [23], the authors built a knowledge pool for reasoning-driven AI-native systems that enable online learning and fast inference of different network functions.

- *Radio Access Networks*: Most new HCI and sensing & communication devices will access to RANs for a better user experience and flexible deployment. As a result, 6G RANs should support massive devices with diverse KPI requirements. To improve user experience in real-time interactions, ISAC is a promising technology that exploits a shared multi-antenna system and advanced signal processing algorithms for data transmission and environment sensing [11]. In addition, a space-air-ground-sea integrated network is promising to enable seamless connectivity for global interactions in the Metaverse [12]. As the tasks and applications in the Metaverse evolve over time, Open-Radio Access Networks (O-RAN) with programmable network functions can reduce the cost for network deployment and upgrades significantly [23].

In summary, based on the above characteristics, we propose the task-oriented design framework in the following, which helps us develop the infrastructure.

- **Human-Computer Interface**: Existing input/output (I/O) systems, such as keyboards, mice, microphones, and monitors, are mainly designed to process texts, audio, and videos. They are inconvenient to support fundamental tasks in Metaverse, such as pose and eye tracking, positioning, haptic control and feedback, and semantic segmentation. Future HCIs should be designed to support these tasks. Thus, we need a task-oriented design. As mentioned in our manuscript, new HCIs, such as virtual reality glasses, tactile internet, and brain-computer interface, will serve as one pillar of infrastructure.

- **Sensing and Communications**: To better support the above tasks in Metaverse, the servers need to acquire information from users and environments promptly. Nevertheless, this could be expensive regarding the costs of sensing devices as well as computation and communication resources. Instead of simply updating 5G KPIs in 6G, we need to deploy and allocate resources according to the task requirements. This is another motivation for task-oriented design. Tasks in the Metaverse have unique sensing and communication needs. Therefore, task-oriented sensing and communication design can help ensure that the sensing and communication technologies used in the Metaverse are well adapted to the tasks that users want to perform. The development of this technology can address key issues such as efficient use of computational resources, improved processing times to alleviate bottlenecks in real-time interactions, and savings in sensor deployment costs, which are critical to achieving an immersive and seamless Metaverse.

- **Network Architecture**: 6G network will bridge the new HCI, sensing & communication infrastructure, and the digital world on the server. Due to the long communication delay, executing all the tasks in a global server is impossible. To address this issue, we need to assign tasks to central servers, local servers, and end-user devices. In other words, we need network functions to support task-level resource allocation and task offloading. Meanwhile, to support massive devices with stringent task-oriented KPIs. From a task-oriented perspective, based on flexibility and user customization requirements, network architecture can be designed to leverage distributed computing, storage, and communication resources in central servers, local servers, and end-user devices. This ensures that the network is optimized for the tasks that users want to perform. The development of this technology can improve network performance and efficiency, making it easier for users to interact with the Metaverse.

1.4 Road Map Towards the Metaverse

In this section, we discuss the road map towards the full vision of the Metaverse as illustrated in Fig. 1.2.

1.4.1 Establish Multi-tier Metaverse in Multi-tier Architecture

The first step toward the Metaverse is to build digital worlds. There are three types of digital worlds: 1) an imaginary environment that does not have a real-world counterpart; 2) the digital twins of a real-world environment; and 3) a digital world overlays on the physical world or even has the ability to reprogram the physical world in real-time. To provide quick responses from the Metaverse to users, we need to use the computing, storage, and communication resources of a local server or end-user device. Then, the states of the user or its digital model are updated to

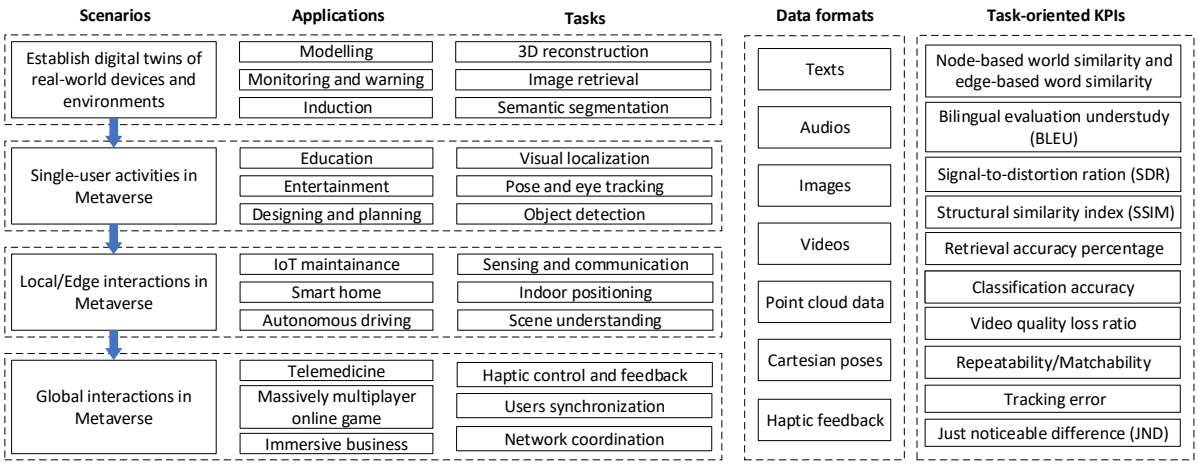


Figure 1.2: Road map towards the full-vision of the Metaverse.

the global server for synchronization. The multi-tier network architecture lays the foundation for building multi-tier digital worlds that support real-time interactions among users from all over the world. Specifically, the raw materials for establishing a Multi-tier Metaverse come from a multi-layered Metaverse virtual world. Depending on the application scenario, the information is distributed in different ways for processing content. The interactive information in the Metaverse can be delivered to the server for processing, while the rendering of the virtual scene can be processed by the local Graphics Processing Unit (GPU) on the user end-device side.

To build the first type of virtual world (e.g., platforms for online gaming), we first create a shared space with virtual roles that are controlled by human users. Then, we develop the rules for information exchange that enable user interactions. For the second type of virtual worlds, we need to simulate the physical and chemical properties of real-world environments (e.g., mass, density, viscosity of fluid, and flammability). In addition, the synchronization between virtual worlds and real-world environments plays a crucial role in real-time decision-making. For example, in factory automation, the digital twins of the real-world factory can be used to monitor ongoing manufacturing processes, predict potential faults, and send warning messages to administrators.

Simultaneously, the digital model of real-world objects is created in the virtual world. The digital model is the prerequisite for achieving all types of interaction modalities. On the one hand, the digital twins require the virtual world requires a shared sense of space, time, object presence, and shared rules for information exchange and user interactions (e.g., avatar, gesture, text, voice, etc.) [24]. On the other hand, the virtual world can reflect the properties of the real world in real-time, including the object's physical and chemical properties (e.g., mass, density, viscosity of the fluid, and flammability) as well as all types of functions. The digital twins (including both environments and objects) can be used to accelerate prototype design, monitoring, decision-making, testing, and repair in industrial applications. Representative applications include

the modeling of virtual environments, digital twins for monitoring and warning offshore energy facilities conditions, and induction and prediction for operating the supply chain.

1.4.2 Single-User Activities in the Metaverse

For single-user activities, all the virtual objects and environments can be built on the end-user device, e.g., a personal computer. With the help of various HCI, the user can interact with everything in the digital world, where several applications in education, entertainment, design, and planning become possible. Specifically, the single user can participate in a variety of single-player activities and enjoy an immersive user experience. Based on the constructed scene, a variety of applications in the Metaverse can be accomplished by relying only on the computing and communication units of the standalone end devices without the assistance of servers. For example, the user can wear clothing with massive sensors and devices (e.g., digital textiles, and exoskeletons) and head-mounted wearable displays or mobile headsets (ultra-high-definition VR glasses). Humans can 1) synchronize their actions with avatars representing users in the virtual world, 2) operate with different virtual objects by different approaches (e.g., virtual touch), and 3) create virtual content (e.g., NFT). Representative applications include immersive entertainment such as Beat Saber [25], immersive teaching and learning, and design and creation in the Metaverse. In summary, by synchronizing users' actions with their digital models in the Metaverse, the users can operate virtual objects and create virtual content (e.g., driving a vehicle or painting).

Nevertheless, establishing a digital world on the end-user device is not easy, as the device has limited computing and storage resources. Thus, the processing delay could be the bottleneck for real-time interactions. In addition, high motion-to-photon values will send conflicting signals to Vestibulo-Ocular Reflex (VOR) and then might cause dizziness or motion sickness. To address this issue, low-complexity 3D reconstruction and segmentation algorithms are in urgent need.

1.4.3 Local Interactions in the Metaverse

In some private networks or local area networks, information is exchanged among devices and users in a small area and is used collaboratively by multiple users and devices without the involvement of the Internet. In these scenarios, all the devices and human users can interact with each other via a local Metaverse. For example, in a smart factory, sensors monitor manufacturing processes and update their states to the local server, where the digital twins of the factory are built [5]. In the digital factory, it is possible to simulate the outcomes of different actions. If an accident is detected in the simulation, the local server sends commands to actuators to stop the processes in an anticipatory manner. Considering the Human-Robotics Collaboration scenario, the information from multiple Internet of Things (IoT) devices and digital twins is used collaboratively to improve productivity and maintenance efficiency. Simultaneously, operators

can also interact with real-world objects and environments with parallel simultaneous actions and states in the virtual worlds. The real-world objects and environment can be enhanced from diversified perceptual degrees (e.g., images, holographic, and voice) via XR technology assists (e.g., holographic see-through glasses, wearable and miniature projectors, and haptic controllers). Another application scenario is the connected car, a local base station can integrate information from multiple vehicles. By reflecting the information to AR/MR devices worn by nearby drivers, pedestrian safety and driver experience can be improved [26]. Since the data are stored and processed on a local server that is not connected to the Internet, this approach can protect users' privacy and avoid security issues.

To enable real-time interaction in the local Metaverse, the latency, reliability, throughput, and coverage of radio access networks are crucial. Although with high-frequency bands, it is possible to improve throughput in a radio access network, wireless signals with high-frequency bands are sensitive to blockages. To achieve ultra-high reliability, multi-connectivity is a promising approach, where backup links can maintain service continuity.

1.4.4 Global Interactions in the Metaverse

The ultimate goal of Metaverse is to support global interactions for a large number of users using different types of HCI. At this stage, remote healthcare, immersive business, and online education will be possible. For example, unlimited numbers of users can access multiple shared and persisted realms via heterogeneous interaction modalities powered by all types of XR technology to collaborate in finishing diverse tasks (e.g., live meetings, live concerts, and live tributes). Users can precept information from both the real world and virtual world, communicate, create, and share virtual content via diverse tools, and interact with multiple users represented by the Avatar in a synchronization way. Asynchrony will lead to inconsistent information received by different users, resulting in inconsistent interaction between each other and affecting user real-time manners. Representative applications include a variety of smart healthcare, massively multiplayer online games and social networking, and immersive business.

With the development of AI technology, Metaverse is becoming more and more reliant on the performance of deep learning models. However, most algorithms work well in small-scale problems. To apply these algorithms in large-scale networks with diverse applications, scalability, and generalizability are bottlenecks. The multi-tier Metaverse built upon the multi-tier network architecture is a potential solution to address the scalability issue. On top of the multi-tier architecture, deep learning or deep reinforcement learning algorithms can be executed in a distributed manner. Further considering the diversity of applications and network topology, the generalization ability of learning algorithms is critical for global interactions in the Metaverse.

In addition, latency remains one of the major issues for global interactions. Specifically, propagation delay is inevitable in long-distance communications. Stochastic network congestion leads to long queueing delays. In addition, the re-transmission scheme in the existing Transmission

Control Protocol/Internet Protocol brings significant latency. Although some interesting ideas have been put forward to achieve real-time interactions [27], the implementation in large-scale networks remains a challenging goal. Based on that, the Tactile Internet is one of the potential key enablers of global user interactions in 6G. The round-trip delay of Tactile Internet should be less than 1 ms to provide a satisfactory user experience. Nevertheless, electromagnetic waves can only travel around 300 km within 1 ms. In addition to the propagation delay, the coding delay is inevitable with traditional source coding. One promising approach to handle coding delay is to optimize the sampling, prediction, and communication systems jointly [27]. By adjusting the sampling rate and prediction horizon by a deep reinforcement learning algorithm, it is possible to improve communication efficiency. Although some interesting ideas have been put forward in this area, hwo to achieve global interactions in the Metaverse still needs further investigation.

1.5 Computer Vision for Metaverse

1.5.1 Computer Vision

As one of the most popular and prominent interdisciplinary fields within artificial intelligence, the expeditious progress of Computer Vision (CV) is actively playing as the building block, and transforming every aspect of the Metaverse. This includes the framework of interpersonal interactions, the IoT, as well as virtual and physical fusion, permeating diverse facets of contemporary society, etc. 6G is recognized as the key enabler that combines with real-time CV and served as the key infrastructure to host a range of emerging applications in the Metaverse e.g., connected vehicles, remote control with avatars. These applications require us to generate timely feedback from the environment through the real-time transmission of data in various CV formats (e.g., images, videos, and point cloud data). However, real-time performance heavily depends on communication networks when sensors and computing resources are separated at different devices or locations. In addition, this gap is widening between the increased demand for communications due to the rapid advancement of CV and the load capacity that can be provided by today's communications. For example, in VR/AR/XR/MR applications, low-latency and high-date-rate communications are required to offload compute-intensive tasks to edge servers [21]. In teleportation applications, where a human operates machines remotely, reliable communications are needed to provide remote viewing for the operation [28]. In the industrial Metaverse, the high-performance synchronization of different entities (including machines, robots, environment, humans, materials, etc.) should be guaranteed for a seamless and safe operation [27]. This has driven us to rethink the technology path that can sustainably meet the growing communication demands of CV for the Metaverse. This includes: (1) investigating various emerging communications-enabled CV applications in the Metaverse. (2) re-examining the demands of CV for specific KPIs in communication systems, (3) re-thinking whether the current communication design framework is compatible and supportive

Table 1.1: CV KPIs [29]–[67]

Self-driving				
Object Detection	Image Segmentation	Pedestrian Detection	Trajectory Prediction	Image Classification
AP/mAP/mAPH /IoU/mIoU	AP/mAP /IoU/mIoU	Median IoU	MSE/MAE /FDE/ADE	OA/F1 Score/ /Accuracy /Precision /Recall
Frame Per Second (FPS), Jitter of FPS, Task Completion Time				
XR Application				
Human Pose Estimation	Image to Image Translation	Human Motion Forecasting	Gaze Estimation	3D Reconstruction
MOTA/AP /mAP/MPJPE /PCK/PCKh	IoU/mIoU /MSE/PSNR /SSIM /MS-SSIM /GAN Loss	MAE/MSE	AP /Gaze Accuracy /Gaze Direction Error /Angular error	F1 Score/IoU /Normal Consistency /Chamfer-L1/ASD-V /ASD-O
Frame Per Second (FPS), Jitter of FPS, Task Completion Time				
Others				
Human-Object Interaction	Medical Image	Anomaly Detection	Real-Time Neural Rendering	Novel View Synthesis
AP/mAP /MPJPE/HOI Classification Accuracy	Dice Similarity Coefficient /Jaccard Index /PSNR/MSE /MAE/ROC /AUC	/Precision /Recall /F1 Score	SSIM/LPIPS/PSNR	IoU/SSIM /MS-SSIM /PSNR/LPIPS /FID
Frame Per Second (FPS), Jitter of FPS, Task Completion Time				

of various potential CV design frameworks in the long term. 4) Review novel methodologies and innovative technologies for the next-generation communication network that supports CV and diverse Metaverse applications.

- *The Gap between CV and Metaverse:* CV plays a critical role in modeling digital avatars and rendering 3D scenes to users in the Metaverse. For example, object Recognition and Segmentation are the basis for understanding and interpreting digital environments, which involves identifying and classifying different objects in the environment, as well as defining their boundaries [69]. Modeling and rendering objects with high variability and complexity requires highly robust and adaptive recognition algorithms. This field advances rapidly due to the evolution of deep learning techniques. However, the existing communication infrastructure does not have sufficient storage, computation, and network resources to support large AI models for real-time CV.

- *CV for Real-Time Applications:* The performance of CV tasks is critical to ensure the overall performance of emerging real-time Metaverse applications, e.g., connected vehicles [70], XR applications with Avatar [71], teleoperation [72], etc. As shown in Table 1.1, the tasks

Table 1.2: Communication KPIs [68]

Use case (high level)		Availability (%)	Cycle time (ms)	Typical payload size (bytes)	Device number	Typical service area
Motion control	Printing machine	>99.9999	<2	20	100	100 m × 100 m × 30 m
	Machine tool	>99.9999	<0.5	50	∼20	15 m × 15 m × 3 m
	Packaging machine	>99.9999	<1	40	∼50	10 m × 5 m × 3 m
Mobile robots	Cooperative motion control	>99.9999	1	40-250	100	<1 km^2
	Video operated remote control	>99.9999	10-100	15-150000	100	<1 km^2
Mobile control panels with safety functions	Assembly robots or milling machines	>99.9999	4-8	40-250	4	10 m × 10 m
	Mobile cranes	>99.9999	12	40-250	2	40 m × 60 m

in connected vehicle applications include semantic segmentation, trajectory prediction, and lane recognition. The performance of these tasks is measured by MSE [73], Peak Signal-to-Noise Ratio (PSNR) [74], and Structural Similarity Index Measure (SSIM) [74]. The MSE and PSNR metrics measure the per-pixel level similarity between two images, and SSIM, on the other hand, measures the structural similarity between them. Average Precision (AP) evaluates the precision in object detection tasks. In XR applications, typical tasks are human position estimation, image-to-image transmission, human motion forecasting, gaze estimation, 3D reconstruction, and rendering, which are measured by another set of KPIs, such as Multiple Object Tracking Accuracy (MOTA) [75], Percentage of Correct Keypoints (PCK) [62], and SSIM [74]. MOTA is a metric used to evaluate the overall accuracy of a multiple object tracking system, various aspects, including false positives, false negatives, identity switches, and mismatches. PCK evaluates the accuracy of keypoint localization by calculating the percentage of correctly predicted key points within a certain distance threshold compared to the ground truth key points. Traditionally, KPIs are used to quantify the performance of specific tasks, such as in connected vehicles, semantic segmentation, trajectory prediction, and lane recognition are required to perform. On the other hand, standardized benchmarking involves evaluating the performance of CV models or algorithms on widely accepted datasets, providing a common evaluation framework that allows researchers and practitioners to compare their methods against others in the field [76]. For instance, benchmarking in Car Learning to Act (CARLA) can be configured to run a number of scenarios that combine different maps, sensors, and weather conditions,

where developers can analyze and compare the performance of their autonomous driving algorithms in a unified and standard way [77].

However, communication systems have not been given much consideration, either at the task level or at the application level. In fact, uncertain communication channels, tight channel resources, and constrained computation and storage space make it necessary to jointly consider sampling, communications, computation, and storage loads [19]. The global balance between different sub-tasks and trade-off between real-time performance and accuracy performance is required for further investigation [78].

- *Communications for Real-Time Applications* It is well known that the 3GPP standard categorizes the communication service into three main branches, i.e.,eMBB, mMTC, and URLLC [9]. The URLLC has stringent requirements on latency (around 1 ms) and reliability (up to 99.99999%). Based on that, more fine-grained investigations on the Quality of Service (QoS) for communication systems in different vertical industries are provided in the 3GPP standard [9]. For example, the scenarios that correspond most directly to autonomous driving defined in the 3GPP standard [9], cooperative motion control that requires > 99.9999 availability, 1ms cycle time (ms), 40-250 bytes typical payload size, 100 device number, and < 1 km^2 typical service area. Nevertheless, it still cannot fulfill all the KPIs in emerging real-time CV applications like self-driving, Virtual/Augmented Reality (VR/AR) [21]. It is expected that in the 6G, the communications can enable KPIs in more diverse dimensions, such as peak data rate, Spectrum Efficiency (SE)/throughput/Energy Efficiency (EE)/network availability/security as well as low Age of Information (AoI)/jitter/round-trip delay [79].

- *The Gap Between the KPIs of Communications and the Requirements of CV in those overlapped scenarios:* Although the 5G standard has defined the communication KPIs for different application scenarios related to real-time CV applications, the accurate requirements of specifically real-time CV sub-tasks on future communications are still ambiguous. The ambiguity is all-encompassing and becomes more urgent to resolve with the rapid iteration of CV technology. In fact, the KPIs of general-purpose communication network architecture is not defined for specific tasks. In other words, the traditional KPIs, e.g., latency, reliability, and throughput, cannot cover the diverse requirements of specific real-time tasks [9]. This misalignment leads to significant challenges in future communication system design. Specifically, the existing communication network design approaches divide the whole system into multiple sub-modules for separate optimization and cannot break the barriers among the sub-modules. As a result, it is difficult to provide E2E performance guarantees for the 6G vision [80]. The impact of network resource allocation on KPIs of real-time CV applications remains unclear, and there is no theoretical model or closed-form expression that can quantify their relationships. To overcome this difficulty, we need to

exploit new KPIs and novel design methodologies. Furthermore, relying solely on communication system design is insufficient to tackle challenges in real-time CV applications, as the overall system performance depends on numerous factors beyond communications. These factors include sensing, sampling, computing (inference and rendering), storage, etc. Thus, it is essential to adopt interdisciplinary approaches to address these challenges, where the distinct characteristics of various real-time CV applications should be considered.

1.5.2 The-State-of-the-Art

Over the past few years, there has been a notable increase in both research and industrial endeavors focused on enabling real-time CV applications. These endeavors encompass various areas, such as network architectures, parallel computing, domain-knowledge-assisted learning, scalability, and generalizability.

Multi-Tier Network Architecture

Multi-tier network architecture with distributed computing capability is believed to be a promising approach. It provides CV applications access to a wide range of shared computing resources in the network (including fog, edge, and cloud), and allows each application to choose the most suitable resource (close to them) to meet its KPI requirements [8]. In the multi-tier architecture, coordination across multiple tasks in core networks is required. The authors in [23] built a knowledge pool for reasoning-driven AI-native systems that enable online learning and fast inference of different network functions and tasks. In addition, new radio access networks are required to improve the user experience in real-time interactions. For example, ISAC is a promising technology that exploits a shared multi-antenna system and advanced signal processing algorithms for data transmission and environment sensing [11]. Space-air-ground-sea integrated networks aim to enable seamless connectivity across different networks [12]. Further considering the evolution of applications in the future, O-RAN offer flexibility in network deployment and upgrades as the network functions are programmable [23].

Parallel Computing Architecture

To improve the real-time performance of CV applications, parallel computing is promising in accelerating the training of neural networks, data analytics, simulations, and visualizations [81]. It has been demonstrated in [17] that the training time can be reduced by 100 times and the rendering can be done in milliseconds by using the small and self-contained

framework [82] and the hash coding. In addition, high-performance real-time computing plays an essential role in building digital twins that incorporate real-time sensing, communications, and control processes [83].

Domain-Knowledge-Assisted Deep Learning

Data-driven AI algorithms are powerful in real-time CV applications, as they do not rely on theoretical models or assumptions. To improve the training efficiency and final performance of AI algorithms, domain knowledge plays a key role in feature engineering, sample selection, value function design, etc [79]. One of the most successful examples is Neural Radiance Fields (NeRF) which uses domain knowledge to enhance its ability to generalize to new viewing angles [84].

Scalibility and Generalizability

Most of the existing deep learning methods work well in small-scale CV problems. As the scale of problems increases, the training/inference time increases rapidly. As shown in [12] scalability remains an open issue in the context of Metaverse applications. To support various tasks in different sensing and communication technologies, deep learning algorithms trained on a data set should maintain good performance in different use cases after a few steps of fine-tuning. The generalization capability is critical and has attracted significant attention from the research community [85] [86].

1.5.3 Future Scope

Considering the wide variety of real-time CV applications and their diverse KPIs, it requires considerable additional research efforts beyond what the communication community has done so far. This proposed special issue aims to bring together leading researchers in both academia and industry with diverse backgrounds to investigate cross-system design methodologies and technologies for real-time CV applications. Thus, we have to

- clarify the CV tasks, communication KPIs, and research challenges in the context of different real-time applications;
- investigate new network architectures and cross-system design methodologies to address these research challenges;
- identify the potential issues and benefits of cross-system design and other related technologies.

The diverse KPIs of real-time CV applications and the complex integration of CV, edge computing, and cloud computing via communication networks create unprecedented research challenges for future communication system design, driving the need for novel and task-specific communication technologies and design methodologies.

Network Architecture – Device, Edge, and Cloud for Real-Time CV

To support real-time CV applications, we need to exploit distributed computing, storage, and communication resources in local devices, edge servers, and central cloud servers. This motivates us to revisit network architecture, where the network functions are predetermined and are not flexible. As KPIs requirements may evolve over time, network functions should be programmable and easily implemented with the available resources of the network. Although huge AI models like ChatGPT are powerful in different tasks, they require a considerable amount of storage and computing resources and can not support real-time CV applications. To overcome this difficulty, designing small AI models that can be implemented at the edge of the network is a promising direction. Meanwhile, the deployment of network resources should be aligned with the requirement of AI models.

The architecture design also needs to incorporate the privacy issue since many CV applications involve privacy-sensitive data, for example, images and videos in hospitals or care homes. Thus, it is important to train AI models in a distributed manner without exchanging raw data, such as Federal Learning (FL), split learning, and multi-tenant learning. Meanwhile, the network architecture design should better support these learning frameworks.

Immersive Technologies – VR/AR/XR/MR for Real-Time Human-Computer Interaction

Supporting VR, AR, XR, and MR for real-time CV applications via wireless communications is becoming increasingly attractive in the 5G and the upcoming 6G era. These applications involve many computing-intensive CV tasks, including complex environmental analysis, hand position estimation, object classification, and image-to-text translation, which are difficult for mobile devices. By offloading those tasks to edge servers with much stronger computing power, the systems can achieve a much better user experience [87]. In particular, when CV applications involve long-distance communications and user interaction/collaboration across different continents, latency becomes a key issue. The predication technique (such as generative AI) is a promising candidate to address the challenge and meet the required CV KPIs, for example, the performance of stable and high frames per second for all users.

Cross-System Design – Task/Goal-Oriented, Semantic, Context-Aware Communications for Real-time CV

Cross-system design is another promising direction in vertical industries, such as robotic automation and MR. Note that cross-system problems are analytically intractable in general. To tackle these problems, we should leverage the advances of data-driven deep learning methods. For example, semantic communications or task/goal-oriented communications focus on completing a specific task rather than conveying information bits.

Nevertheless, a lot of technical issues in cross-system design remain open. Specifically, training deep neural networks for Joint Source and Channel Coding (JSCC) or other types of semantic-aware and task-oriented communications requires a large amount of data samples and computing resources, which are not feasible for mobile devices. Another challenge is the generalization of the cross-system design. For example, in task-specific update policy design, AI models usually need to be re-trained to handle new tasks.

Applications in Vertical Industries

Real-time CV lays the foundation for robotics and unmanned vehicles as it allows computers or human operators to understand the environment and make decisions in real-time. For example, in autonomous driving for future transportation, edge inference is a promising technique to achieve excellent inference performance – including scene segmentation, pedestrian detection, and trajectory prediction – via strong computer power at edge servers and multiple data streams collected from different vehicles. Here, joint optimization and trade-off across communications, computing, and real-time CV tasks are essential for performance assurance, where large latency (delayed inference) has significant impacts on driving safety.

In Simultaneous Localization And Mapping (SLAM) for extreme environments, including mining, rescue and search, and nuclear decommissioning applications, multiple unmanned aerial/ground vehicles (also called multi-agents) need to exchange/share images and other sensing data for coordinated mapping and control. Poor communications and resource management would significantly slow the mapping process.

In applications of digital twins for future manufacturing, for example, safety monitoring and management, the system uses multiple cameras to detect and track objects (humans, vehicles, materials, and machines) in real-time, analyze their relationship in the digital twins, and activate alerts when needed. Here, the delayed CV outputs will cause asynchronization between real-world objects and their representations in the digital twins and may lead to the failure of the whole monitoring system.

1.6 Challenges of Task-Oriented Cross-System Design for the Metaverse

1.6.1 Data Structure

The data structure of a task depends on the HCI or environment sensing technologies. Traditional speech and image signals are represented by time-series data and the red-green-blue (RGB) model, respectively. Nevertheless, due to the diverse human-computer interaction modes in the Metaverse, the data structure has also changed from being characterized by the existing standardization gird measurements by calculating 1D-2D Euclidean distance to utilizing stereoscopic and multidimensional features like graph signals. For example, spatial correlation is critical for tactile signals and brainwave signals and relies on the topology of the sensors. The topology information is useful in signal processing and may facilitate the execution of tasks. In addition, the signals generated by a radar system or depth-sensing camera, such as point-cloud data are converted into 3D tensors in the Euclidean space before they can be processed by convolutional neural networks. This procedure causes additional computational overhead and processing delay. To reduce overhead and improve the performance of a task, the authors of [88] developed a PointNet to handle a range of tasks in environment sensing, such as 3D shape classification and segmentation. Nevertheless, a widely accepted standard for data storage, processing, or communications is still missing in the Metaverse, and it will lay the foundation for immersive interactions among human users, machine-type devices, and environments. Thus, as the great enabler for enabling human-computer interaction, data structures face challenges in the Metaverse: (1) How to characterize the features of different data for different applications in the Metaverse, and (2) How to design the frame structure to cope with a variety of tasks that need further investigation.

1.6.2 Task-Oriented KPI

Diverse tasks in the Metaverse have stringent requirements on a range of KPIs, which are still difficult to fulfill. For example, in image transmission tasks, an image is first compressed at the transmitter side, then transmitted over the communication system, and finally reconstructed by the receiver. The SSIM is used to quantify the image quality degradation caused by these steps. In 3D reconstruction tasks, point-cloud data repeatability and matchability are the two commonly used task-oriented KPIs for measuring the similarity of 3D structures. Another example is in applications that require low-latency feedback, the user-experienced delay should be close to zero, but the propagation delay could be up to dozens of milliseconds when the communication distance is hundreds or thousands of kilometers. Furthermore, the KPIs defined in the 5G standard, such as throughput, latency, and reliability, are not the same as the task-oriented KPIs as illustrated in Fig. 1.2. For instance, in haptic communications, it is natural to raise a question: Do we really need to guarantee the 99.999% reliability in communication systems in order to achieve the target JND?

There is no widely accepted standard for processing tactile and brainwave signals. The impact of network resource allocation on task-oriented KPIs remains unclear, and there is no theoretical model or closed-form expression that can quantify their relationships. To overcome this difficulty, we need novel design methodologies.

1.6.3 Multi-Task Processing and Coordination

With the multi-tier network architecture, tasks of an application may be executed by the end-user device, an edge/local server, or the cloud server. The offloading and coordination of multiple tasks are not trivial since they are interdependent. Thus, the interdependence and causality between tasks implicate tasks should be complicated sequentially. For example, in the tactile internet, the task of perception and understanding as the goal motivation should be performed first, before completing the task of generating feedback signals to the wearable device. Offloading tasks to the edge or cloud provides a flexible solution to alleviate the communications and computation constraints on the end-user devices. However, different tasks have different sequences, deadlines, and priorities. Therefore, how to offload tasks and coordinate end-user devices, local servers, and global servers to generate quick response scenarios is still an open question. For some highly interactive applications in the Metaverse, the end-user device senses the behavior of the user and then communicates with the local server, where the feedback is generated. Finally, the local servers synchronize the states of users in a cloud server. Delays or packet losses in any of the tasks will have a serious impact on the overall performance of the application. To provide a satisfactory user experience in the Metaverse, we need to break the barriers among sensing, communications, and computing systems, and jointly design the whole network.

1.6.4 Privacy, Security, and Trust

In the Metaverse, diverse HCI, in-depth use of AI, and high-frequency data exchange also raise critical issues on privacy, security, and trust [89]. For example, scene reconstruction requires access to the environment of the room. Brain-computer interface-driven tasks need to measure and process human brainwave signals. As a result, traditional centralized and single-module authentication, and data storage, and is no longer sufficient to meet the security challenges. Decentralized learning algorithms that do not need to share raw data among users and servers are promising. Even worse, most cross-system designs do not offer a performance guarantee in terms of classification or regression accuracy. In addition, to provide more realistic scenes and a more intelligent mechanism that facilitates task performance, intensive use of AI requires a variety of high-precision privacy data collected from HCI. One potential solution is blockchain technology. Blockchain technology provides a promising encryption and distribution solution. Based on that, the Metaverse can be governed by community-led entities known as decentralized autonomous organizations (DAOs), where all participants have the decision-making power to manage their

assets. However, hacking incidents still occur due to the low level of standardization and the continued lack of transparency. More diverse data encryption methods, data sharing protocols, and cross-domain security and trustworthiness systems need investigation.

1.7 Potential Solutions

1.7.1 Cross-System Design

Existing HCI, sensing, communications, and computing systems are developed separately. This design approach leads to sub-optimal solutions, brings extra communication overhead for coordinating multiple tasks, and can hardly meet the task-oriented KPIs. To address these issues, a cross-system design has been investigated in the existing literature. There are several existing cross-system design approaches. (1) As shown in [22], when dealing with reconstruction tasks including in-text sentences, sounds, images, and point cloud data, by joint source and channel coding, it is possible to achieve a better quality of service at low signal-to-noise ratios. Nevertheless, complicated coding schemes may bring extra processing delay, which remains an issue in ultra-low latency communications. (2) Considering the cost of deploying a large number of sensors, integrating sensing into communication systems is a promising approach, as cellular networks have been widely deployed [11]. By utilizing communication signals in environmental sensing, cellular networks can support a variety of tasks, such as localization, object detection, and health monitoring. (3) Given the fact that state observations are outdated in some tasks, the Metaverse needs to respond to users' actions in an anticipatory manner. To achieve this goal, prediction and communication co-design is promising, especially for applications in the Tactile Internet that require ultra-low latency [27]. It is worth noting that cross-system problems are, in general, very complicated and may not have well-established models. As a result, most of the existing analytical tools and optimization algorithms are not applicable.

1.7.2 Domain-Knowledge-Assisted Deep Learning

To solve the above cross-system design problems, data-driven deep learning methods are promising, as they do not rely on theoretical models or assumptions. However, straightforward applications of deep learning may not generalize well with diverse task-oriented KPIs and data structures [90]. To address this issue, one should exploit domain knowledge in feature engineering, sample selection, value function design, etc. When deep learning is adopted in task-oriented design, there are three major issues. (1) Most of the existing deep neural networks work well in small-scale problems. As the scale of the problem increases, the training/inference time increases rapidly. In the Metaverse, there could be millions or billions of users and devices, and thus scalability remains an open issue. (2) To support various tasks in different sensing and communication environments, deep learning algorithms trained on a data set should achieve good performance

in different use cases after a few steps of fine-tuning. This generalization ability is critical for using deep learning in the Metaverse. (3) Most deep learning algorithms do not offer a performance guarantee in terms of classification or regression accuracy. But the KPIs required by some mission-critical tasks are sensitive to the outcomes of learning algorithms. Improving the safety of deep/reinforcement learning algorithms by exploiting domain knowledge is a promising and vital approach.

1.7.3 Universal Design

The Metaverse aims to provide better interactions among users with different cultural backgrounds and health conditions (e.g., careers, nationalities, abilities or disabilities, etc.). Different users may have different preferences, habits, and cognition. Meanwhile, they may use different types of HCI devices with different data structures. The diversity of users brings significant challenges in the design of the Metaverse, and the universal design is essential for the success of the Metaverse by considering the diverse needs and abilities of all the users throughout the design process, standardization, and government regulation. For example, a universal design platform named Omniverse can meet user demands from different backgrounds (e.g., artists, developers, and enterprises), whereas Universal Scene Description is promising to be the open and extensible standard language for the 3D Internet to eliminate barriers among different user communities [91]. Nevertheless, a lot of effort is still needed in the universal design.

In summary, cross-system design is promising to solve the challenges of task-oriented KPIs and multi-task processing and coordination. Diverse tasks in the Metaverse have stringent requirements on various KPIs, which are difficult to fulfill. Existing HCI, sensing, communications, and computing systems are developed separately. This design approach leads to sub-optimal solutions and brings extra communication overhead for processing and coordinating multiple tasks. By cross-system design, the interactions between different systems can be exploited to handle multi-task processing and coordination, thus guaranteeing the KPIs at the task level.

By exploiting domain knowledge on the HCI and sensing technologies, domain knowledge-assisted deep learning has the potential to meet task-oriented KPIs for tasks with diverse data structures. Straightforward applications of existing deep neural networks may not be generalized well for different data structures from different HCI and sensing technologies, such as wearable tactile devices, brain-computer interfaces, and joint sensing and communication technologies. To guarantee task-oriented KPIs for diverse data structures, deep neural networks should have strong generalization ability. We propose to use domain knowledge for feature engineering and hyper-parameter selections in deep learning algorithms.

Universal design can address the challenge of data structures and task-oriented KPIs by considering users' diverse requests and abilities throughout the design process, standardization, and government regulation. For example, in the context of HCI or environment sensing technologies,

universal design can ensure that the network functions are compatible with a wide range of HCI devices, especially for users with disabilities.

1.8 Future Work and Open Issues

- *Scalability and Generalization Ability*: Most of the task-oriented design methods work well in small-scale problems. As the scale of the problem increases, the computation overhead and processing time in both training and inference increase rapidly. In the Metaverse, there could be millions or billions of users and devices, and thus scalability remains an open issue [12]. In addition, to support various tasks in different sensing and communication environments, deep learning algorithms trained on a data set should achieve good performance in different use cases after a few steps of fine-tuning [79]. This generalization ability is critical for using deep learning in the Metaverse.

- *Green Communications*: The Metaverse in 6G will require high levels of energy to power the network and the devices used to access it [92]. This makes it difficult to rely solely on renewable energy sources and requires a more complex approach to energy management. In addition, The Metaverse in 6G will have a complex network architecture that spans multiple layers, from edge computing to core networks. Designing an energy-efficient network that covers all of these layers is a significant challenge.

- *Privacy, Security, and Trust*: Most task-oriented design methods do not offer a performance guarantee in terms of classification or regression accuracy. But these KPIs required by some mission-critical tasks are sensitive to the outcomes of learning algorithms. In the Metaverse, diverse HCI, in-depth use of AI, and high-frequency data exchange also raise critical issues on privacy, security, and trust [89]. For example, scene reconstruction requires access to the environment of the room and the brain-computer-interface participated tasks need to measure and process human brainwave signals. As a result, traditional centralized and single-module authentication, data storage, and is no longer sufficient to meet security challenges [93]. More diverse data encryption methods, data sharing protocols, and cross-domain security and trustworthiness systems need investigation.

Chapter 2

Related Work

2.1 Task-Oriented Cross-System Design

The task-oriented cross-system design consists of different components including sensing, sampling, compressing, feature extraction, reconstruction, task execution, etc. The task-oriented KPI can be seen as feedback that can guide the design of these modules in real-time. Each model takes the input information and task-oriented KPI as the basis to dynamically adjust the parameters in the corresponding degrees of freedom. Thus, the task-oriented cross-system design framework can benefit from two aspects. On the one hand, the gap between the task-oriented KPI and the low-level KPI (e.g., throughput, latency, reliability) provides the opportunity to save communication resources overhead. Meanwhile, the tasks KPI in the Metaverse provide customized demands that provide a more satisfying and diverse customized user experience.

2.1.1 Sensing and Sampling

Excessive and unnecessary sensing data in the Metaverse can cause a waste of energy and communication resources. Based on the task-oriented customization requirements, we can reduce the communication overhead by designing sensing and sampling strategies. The AoI-based update policy can be considered as the pioneering approach by connecting task performance with the time elapsed by the information generating at the source [94]. However, AoI is an intermediate performance metric that cannot fully capture the specific requirements of the task and is unaware of the burstiness of the source. For example, when the state of an environment or device is stationary, there is no need to update the state frequently. When the state changes rapidly, the source should generate packets and update frequently. Thus, more straightforward KPIs that quantified the packet importance from the source like MSE between the source and destination are often used in the design of task-oriented communication systems. However, these works often assume that the source generation follows specific distributions and expect that these distributions are sufficiently complete to characterize the source characters [95]. Nevertheless,

these assumptions are often difficult to sustain in complex application scenarios of the Metaverse. Based on that, data-driven E2E reinforcement learning is expected to become a mainstream approach, especially in the background of the large amount of data present and the involvement of multiple users and intelligence.

2.1.2 Sampling and Feature Extraction

Sampling here refers to the communication system selecting the keyframes (e.g., words, pictures, point-cloud data) and discarding the others that are not important for the tasks on the time scale. Meanwhile, for each keyframe, feature extraction can be performed, which refers to detecting and compressing the corresponding transmission format by extracting key information and filtering out irrelevant or unimportant information for the task. Therefore, The transmitter can be designed from the task-oriented perspective in both time and space dimensions, transmitting only selected and compressed data thus saving energy and communication resource overhead. Benefiting from the advancement of deep learning especially the popularity of CV, a more intelligent and precise to extract feature information approach can be performed by using various SOTA encoder-decoder models (GAN, VAE, ViT diffusion model). In addition, The development of interpretability of machine learning including the principle of information bottleneck also brings significant guidance on the extraction of key information [96]. With the guidance of task-oriented KPIs, different generators and parameters are designed for information extraction.

2.1.3 Reconstruction

Reconstruction refers to interpreting and restoring data to a state where the task can be completed. It is worth noting that the goal of data reconstruction is not to be as consistent as possible with the data generated by the source but to satisfy the task. This is particularly important for the Metaverse, where the computational and communication load required to reconstruct each sent data from each user in its original form is difficult to estimate and therefore often infeasible. In addition to the encoder-decoder-based model mentioned in the information extraction, neural rendering is also seen as a promising reconstruction method. The potential of this method to reduce communication resources compared to traditional methods based on radar point cloud data scanning is evident. Dozens of multi-angle 2D images can be modeled in a few seconds to produce a detailed 3D model [17].

2.1.4 Sampling-Communication Co-Design

AoI is a performance metric widely used in co-design communication systems and sampling policies (also called state update policies) [97]–[99]. In [97], the authors optimized the sensing and updating policy for an air pollution monitoring application by minimizing the weighted

sum of the AoI and the total energy consumption of the device. By adjusting the weighting coefficients of the AoI and tuning the energy consumption manually, it is possible to achieve the target trade-off. To collect new data from power-constrained sensors in an Industrial Internet of Things (IIoT) network, the authors of [98] optimized a scheduling algorithm by decoupling the multi-sensor problem into single-sensor problems. In [99], the authors considered a status update problem over an error-prone wireless channel, where the average cost of sampling and communications was minimized subject to average AoI constraints.

Since AoI is an intermediate performance metric that does not fully capture the requirement of a specific task and is not aware of the burstiness of the source, it is not a good metric [100]. For example, when the state of an environment or device is stationary, there is no need to update the state frequently. When the state changes rapidly, the source should generate packets and update frequently. For this reason, different performance metrics and design frameworks have been considered to improve sampling efficiency, e.g., goal-oriented communications [101] and mutual information [102]. The authors of [101] developed a goal-oriented sampling and communication policy for status updates over an unreliable wireless channel, where only effective samplings for lowing real-time reconstruction errors were allowed to be transmitted to the actuator. The results show that the proposed strategy can significantly improve effective updates and reduce the cost of actuation errors. [102], instead, used the mutual information between the real-time source values and the samples delivered to the receiver to optimize the sampling policy, proposing a transmitter that maximizes the expected mutual information by sending a new packet once the latter is below a threshold.

2.1.5 Prediction-Communication Co-Design

Prediction plays an essential role in reducing the user-experienced delay in URLLC. To reduce round-trip delay in a VR application, the authors of [103] proposed to predict, pre-render, and cache VR videos in an edge server, where Long Short-Term Memory (LSTM) and Multi-Layer Perception (MLP) neural networks predict body and head motion, respectively. In [104], the authors considered an AR robotic telesurgical application, where, with the help of prediction, they could reduce the task completion time by 19% without increasing the manipulation error rate. The authors of [105] jointly optimized the communications and packetized predictive control system to minimize wireless resource consumption under the control outage probability constraint. The results in [106] indicated that prediction and communication co-design could achieve a better trade-off between reliability and latency than traditional communication systems without prediction. The authors in [107] proposed a task-oriented prediction and communication co-design framework in a haptic control scenario, where the goal is to minimize the required radio resources subject to the low-latency and high-reliability requirements of various tasks. Particularly, the just noticeable difference (JND) is used as the task-oriented performance metric system [107].

2.2 Multi-Tier Computing Architecture

To support diverse applications in Metaverse, a sequence of interdependent tasks should be executed by some deadlines. In addition, task offloading is critical in the multi-tier network architecture [12]. The 6G network should be capable of large-scale multi-tier synchronization to enable the completion of diverse and resource-intensive KPIs in the Metaverse by leveraging distributed computing, storage, and communication resources in local devices, edge servers, and central servers [8]. Thus, 6G core networks need new network functions for task-level resource management and task offloading.

2.2.1 Core Networks

The existing core networks manage resources and quality-of-service at the application level. The session management function will create a new protocol data unit session when there is a new service request. (1) *Semantic-effectiveness (SE) Plane*: To achieve ubiquitous connectivity and support task-level KPIs in Metaverse, the core network should handle the semantics covering the transmitted bits in a given application and use case. The authors of [108] introduced a semantic-effectiveness (SE) plane as a core component of forthcoming communication architectures. This SE plane enhances the protocol stack by offering standardized interfaces that allow for information filtering and direct control of the protocol stack. (2) *Reasoning-driven AI-native Network*: To achieve a sustainable and self-sufficient network in 6G, the authors of [23] introduced the reasoning-driven AI-native systems, where the network can continuously accumulate knowledge and use it to perform operations that would not be possible through excessive reliance on existing data and re-training mechanisms. Thus, future 6G networks are able to leverage causality and stochasticity in the data, discern underlying structures, infer logical relationships, and mitigate semantic noise. (3) The communication network can be abstracted as a neural network with noise and measured with the mutual information metric at the source and receiver [109]. This assumption is not always feasible in practice in the Metaverse where there are various heterogeneous networks and the network performance is not always constant due to the user's decisions. The dynamic network system and the large number of users make the training of JSCC significantly more difficult. Nevertheless, for scenarios where multiple people are online in the Metaverse, scheduling based on a task-oriented communication system can significantly optimize network performance and reduce network blocking. The user-centered network design implicitly starts with various task-oriented metrics defined by users to design the network's queuing policies software-defined network, or network slicing.

2.2.2 Radio Access Networks

Most of the new HCI and sensing & communication devices will be connected to RANs for a better user experience and flexible deployment. As a result, RANs should support massive devices with stringent KPI requirements. Nevertheless, 5G New Radio does not cover this scenario, where interference may become the bottleneck. Furthermore, 6G RANs should support ISAC, which may bring new challenges and opportunities to positioning enhancement, beam management, and channel feedback.

(1) *Integrated Sensing and communications (ISAC)*: Complex environmental perception and inference and ubiquitous communications make the ISAC that combined perception and communications become one of the potential enablers in the future [11]. By simultaneously utilizing data for conveying information and sensing environment information, it can save communications, computing, and storage resources, and achieve a more reasonable resource allocation balance between communications and sensing. This scope, in particular, allows ISAC to be deployed to serve the applications in the Metaverse. For instance, ISAC can enable real-time tracking of user movements, enhancing immersion. Additionally, ISAC-related technology such as massive MIMO and intelligent reflecting surfaces can provide ultra-high bandwidth for Metaverse VR and AR applications [110].

(2) *Space-Air-Ground-Sea Integrated Network (SAGSIN)*: To achieve a seamless and consistent user experience within the Metaverse, it is imperative to have complete coverage that enables global sensing and uninterrupted connectivity. SAGSIN deeply integrates space (e.g., satellites), air (e.g., balloons, drones, unmanned aerial vehicles (UAVs)), ground (cellular/WiFi/wired network), and sea layers, which can enable seamless connectivity across different parts of the Metaverse, regardless of their locations or environments [111]. Besides, it can enable Metaverse to leverage advanced network capabilities, such as edge computing, which can improve the performance and responsiveness of Metaverse applications and services by processing data closer to the user.

(3) *Open-Radio Access Network (O-RAN)*: O-RAN can enable more efficient and cost-effective deployment of wireless infrastructure, which is critical for providing reliable and high-speed connectivity to support the requirements of the Metaverse [112]. The technology allows network operators to use open interfaces and modular components from different vendors, facilitating easier network upgrades and scaling. In addition, the interoperability offered by O-RAN technology can also enable seamless connectivity across different network types and domains, such as cellular, Wi-Fi, satellite, and edge computing, which is important for providing ubiquitous and consistent user experiences in the Metaverse [113].

2.3 Key Enabler for Metaverse

2.3.1 Digital Twins

In Digital Twins applications for future manufacturing, such as safety monitoring and management, the system uses multiple cameras to detect and track objects (humans, vehicles, materials, and machines) in real-time [114]. Then, learning algorithms can be used to analyze their relationship in the digital twin and generate safety messages. The delayed outputs will lead to a mismatch between real-world objects and their representations in the Metaverse, which would result in the failure of the whole monitoring system [115]. However, there is still a research gap on how to exchange and synchronize data between digital twins and Metaverse platforms. This includes data formats, protocols, and methods for real-time updates, ensuring the virtual representations in the Metaverse accurately reflect their real-world counterparts [114]. In addition, diverse communications, computing, and storage demands from different Metaverse applications require refined resource allocation, and how to ensure scalability remains an open issue [116].

2.3.2 Extended Reality (XR)

: Supporting XR for Metaverse applications via wireless communication is becoming increasingly attractive in the 5G and the upcoming 6G era. These applications involve computing-intensive Metaverse tasks, such as complex environmental analysis, hand position estimation, object classification, and image-to-text translation. Due to limited processing resources and battery capacity, it is difficult to execute these tasks on mobile devices. By offloading those tasks to edge servers with much stronger computing power, the systems can achieve a much better user experience [87]. However, despite 6G's promise of ultra-low latency, achieving real-time XR interactions within the Metaverse requires further exploration [117]. Identifying and mitigating latency bottlenecks in XR data transmission and processing is vital to avoid disorienting user experiences. In addition, as XR devices capture and process sensitive user data, ensuring data privacy and security becomes a pressing issue. Comprehensive research is required to develop robust security measures that safeguard user data and identities within the Metaverse [118].

2.3.3 Semantic, Task-Oriented Communications for the Metaverse

The communication paradigm can shift from guaranteeing the correct reception of each single transmitted bit, irrespective of the meaning conveyed by the transmitted bits, to focusing on the transmission bits that are key to accomplishing the goals or tasks by analyzing the semantic context or the relevant information [23]. One of the commonly employed design strategies in semantic communication is the JSCC strategy, which involves the simultaneous training of source and channel coding components. Recent research indicates that JSCC demonstrates superior performance in handling reconstruction transmission tasks across various media, encompassing

in-text sentences, sounds, images, and point cloud data. Notably, it exhibits significant advantages, particularly in scenarios characterized by low signal-to-noise ratios [109]. However, Metaverse is expected to handle large amounts of heterogeneous data from various sources and formats. How to develop semantic data representation and ontology models that enable efficient data exchange, integration, and interpretation within the Metaverse context still needs further investigation [119].

2.4 Timely and Accurate Modeling for the Metaverse

The concept of the Metaverse was initially introduced in Neil's book, *Snow Crash* [2], coinciding with the development of virtual physical fusion technology. Notable pioneering contributions have been made in various applications of the Metaverse, including the game networking [120], autonomous vehicles [121], IoT devices [122], and education [123]. In [120], the authors discussed the features and challenges of real-time online gaming in the Metaverse, where networking technologies lay the foundation for immersive and real-time interactions. In the vehicular Metaverse, the authors of [121] proposed an auction-based mechanism to allocate physical and virtual entities in the synchronization service market, where a synchronization scoring rule and a price scaling factor are employed to improve the total score and protect participants from adverse selection. In [122], IoT devices were deployed to sense and collect status information of physical systems for establishing digital twins that mirror their real-world counterparts in real-time. The Metaverse research agenda for advanced education was overviewed in [123], where enabling technologies, application scenarios, and core ethical considerations are discussed. The above work has yielded valuable insights that could be leveraged to advance the development of the Metaverse, particularly with regard to the conception and refinement of forthcoming communication systems and network architectures.

2.4.1 Resource Management for the Metaverse

The communications, computing, and storage resources in existing IT and networking infrastructures are insufficient to support emerging tasks in the Metaverse. A branch of existing studies strove to address the challenges in task scheduling and resource allocation [124]–[126]. Considering that the arrival and departure processes of applications in the Metaverse are stochastic, the authors of [124] formulated a semi-Markovian decision process and optimized the resource allocation strategy to maximize resource utilization efficiency. In [125], the authors proposed a novel decentralized Metaverse framework, where sub-Metaverses are deployed on mobile edge computing servers. To minimize synchronization time, an iterative algorithm was developed for computing and communication resource allocation. To incorporate federated learning into industrial Metaverse, the authors of [126] developed a high-performance and efficient system to address the learning forgetting issue in scenarios with non-independent and identically distributed data and limited communication bandwidth. The above work revealed some fundamental tradeoffs

between resource utilization efficiency and KPIs in the Metaverse. Nevertheless, finding the optimal policy in the Metaverse remains a challenging issue due to its high complexity.

2.4.2 Task-Oriented Cross-System Design for the Metaverse

To fill the gap between communication KPIs and the KPI requirements of specific tasks, task-oriented cross-system design is a promising approach. The authors of [127] considered an Age-of-Loop metric for the remote control of autonomous guided vehicles and proposed a goal-oriented wireless solution that adjusts the data rate to achieve high control accuracy. Their results showed that with goal-oriented KPI, it is possible to achieve higher accuracy than commonly used communication KPIs, such as Age-of-Information. In [96], the authors proposed a learning-based communication scheme that optimizes feature extraction, source coding, and channel coding in a task-oriented manner to achieve low-latency inference for image classification. The experimental results of this work indicated that task-oriented communications achieve a better rate-distortion tradeoff than baseline methods. More recently, the authors of [128] developed E2E task-oriented resource management by integrating sensing, computing, and communication processes into a joint design framework, where the artificial intelligence model is split and executed on edge servers for low-latency intelligent services. To improve user experience in immersive Metaverse applications, the authors of [129] proposed a user-centered joint optimization approach to optimize frame generation location, transmission power, and channel access arrangement. These studies indicated that by task-oriented cross-system design, it is possible to provide a better user experience and achieve higher resource utilization efficiency. Note that the cross-system problems are non-convex or NP-hard in general. Finding a near-optimal resource management solution with low complexity remains a challenging issue.

2.4.3 Comparison of Existing Work in VR/AR/XR/MR

Supporting VR/AR/XR/MR for Metaverse applications via wireless communications is becoming increasingly attractive in the 5G and the upcoming 6G era. These applications involve computing-intensive Metaverse tasks, such as complex environmental analysis, hand position estimation, object classification, and image-to-text translation. Due to limited processing resources and battery capacity, it is difficult to execute these tasks on mobile devices. By offloading those tasks to edge servers with much stronger computing power, the systems can achieve a much better user experience [87]. However, despite 6G's promise of ultra-low latency, achieving real-time XR interactions within the Metaverse requires further exploration [117]. Identifying and mitigating latency bottlenecks in XR data transmission and processing is vital to avoid disorienting user experiences. In addition, as XR devices capture and process sensitive user data, ensuring data privacy and security becomes a pressing issue. Comprehensive research is required to develop

robust security measures that safeguard user data and identities within the Metaverse [118]. Compared with AR/VR applications, the novelty of the Metaverse is summarized as follows,

- **Large Scale**: The Metaverse aims to be a shared virtual world that encompasses a large number of users and environments, while AR/VR applications are typically more focused on specific use cases or applications.
- **Integration with the Physical World**: The Metaverse aims to seamlessly integrate virtual and physical worlds, providing a more seamless and immersive experience for users. In contrast, AR/VR applications are typically limited to virtual environments.
- **Decentralization**: The Metaverse is being designed as a decentralized network, relying on peer-to-peer networks and blockchain technology to provide secure and transparent interactions. This is a departure from the centralized, server-based architecture that is commonly used for AR/VR applications.
- **Economics**: The Metaverse is being designed as a platform for digital commerce, enabling users to buy, sell, and trade virtual goods and experiences. This is a departure from the more limited economic models that are used in most AR/VR applications.
- **Interoperability**: The Metaverse is being designed as an open and interoperable platform that allows for the seamless exchange of virtual assets and experiences between different virtual environments. This is a departure from the closed, proprietary ecosystems that are common in AR/VR applications.

2.4.4 Comparison of Existing Works in Digital Twins

The Digital Twins concept dates back to NASA's Apollo program in 1970 when NASA built a complete, high-caliber ground-based semi-physical simulation system to train astronauts and controllers, including the simulation of a wide range of failure scenarios [130]. In 2003, the Digital Twins were explicitly used by Grieves in his course on "product life cycle management," which defines it by physical product, virtual product, and their connection [131]. In 2012, NASA clarified the concept of Digital Twins and defined it as "integrated multiphysical, multiscale, probabilistic simulations of an as-built vehicle or system using the best available physical models, sensor updates, and historical data" [132]. In the white paper on the Cyber-Physical Infrastructure from the UK government, the digital twins are depicted as "Digital twins typically start life as simulations or emulations, often modeling a real-life process before it exists." [133]. Basically, the digital twins are regarded as the "mirror" of the physical objects, where the data can be directionally or bidirectionally updated across virtual space and physical space for Digital Twins. Particularly, the digital models refer to the case where there is no update across the two spaces

and use digital shadows to refer to the case where there is a one-directional update from the physical space to the virtual space, as specified in [114].

Significant contributions have been made in the field of digital twins powered by 5G/6G. The authors in [134] proposed a deep reinforcement learning approach to solve the placement and migration problems of digital twins assisted by edge computing while minimizing the synchronization delay between digital twins and physical objects. The work in [122] proposed a resource allocation algorithm for synchronizing IoT devices with their digital models in the Metaverse by using a game-theoretic framework. The authors in [135] proposed an edge continual learning framework that can accurately synchronize a physical object with evolving affinity with its digital twin. In particular, the digital twin is modeled as a Deep Neural Network (DNN) at the wireless network edge to model an autonomous vehicle traversing a dynamic environment. Instead of using a centralized framework, the authors of [122] proposed a framework that jointly synchronizes Digital Twins and sub-Metaverses in a distributed Metaverse framework. Nevertheless, diverse communications, computing, and storage demands from different Metaverse applications require a refined resource allocation. Finding the optimal policy in the Metaverse and ensuring scalability remains a challenging issue due to its high complexity. We have compared our work with existing related works in Table 2.1.

Table 2.1: Existing Works on Digital Twins

Existing Works	Cross-system Design	Real-world Prototype Implementation	User-Centralized Metaverse	Task-oriented KPI	Improved DRL Algorithm with Domain Knowledge	E2E Training
[134]			✓		✓	
[122]	✓		✓			
[136]				✓	✓	✓
[135]	✓					
[137]		✓				✓
[138]	✓			✓	✓	
[139]	✓	✓		✓		✓
[125]			✓		✓	
Our work	✓	✓	✓	✓	✓	✓

2.5 Summary of Contributions

In this thesis, we introduce the task-oriented cross-system design for the Metaverse in the 6G era. we aim to address the following fundamental issues: 1) What are the research gaps between the 5G/6G networks and the Metaverse? 2) How to eliminate the gap between traditional communication KPIs defined in the 5G standard and task-oriented KPIs in the Metaverse? 3) How to implement task-oriented cross-system design in the Metaverse, including sampling, communications, prediction, control, and rendering? 4) How to utilize task-oriented cross-system

domain knowledge to guide the E2E training of Deep Reinforcement Learning (DRL) algorithms? The main contributions of this work are summarized as follows:

- We holistically illustrate the three infrastructure pillars that the Metaverse will be built upon, and depict the roadmap toward the full vision of the Metaverse. We comprehensively review the challenges of task-oriented design in the Metaverse; To tackle these challenges, we put forward potential solutions from a system-level perspective. We also review the existing related work from different perspectives including task-oriented cross-system design, multi-tier network architecture, as well as timely and accurate modeling in the Metaverse.

- We establish a task-oriented cross-system design for a simple case, where sampling, communications, and prediction modules are jointly optimized for the synchronization of the real-world devices and digital model in the Metaverse. Specifically, the sampling rate and the prediction horizon are jointly optimized to minimize the communication load subject to a MSE constraint. To solve the optimization problem, we propose a Knowledge-assisted Constrained Twin-Delayed Deep Deterministic (KC-TD3) algorithm by combining DRL techniques with expert knowledge of sampling, communications, and prediction. Specifically, the following learning techniques are applied to improve the DRL algorithm: 1) extension of double Q-Learning, 2) state-space reduction, 3) interdependent action normalization, and 4) accelerated primal-dual policy optimization (APDO).

- We establish a task-oriented cross-system design framework for a general case, where sensing, communications, prediction, control, and rendering are jointly considered for modeling a robotic arm in the Metaverse. The scheduling policy and the prediction horizon are jointly optimized to minimize the required packet rate to guarantee a modeling error constraint. We propose a C-PPO algorithm by integrating domain knowledge into the advanced Proximal Policy Optimization (PPO) algorithm. Specifically, 1) the Jacobian matrix, which is widely used for analyzing the motion of robotic arms, is included in the state of DRL to improve the training efficiency of C-PPO. 2) The Conditional Value-at-Risk (CVaR) of the state-value function that characterizes the long-term modeling error is applied to formulate the constraint. 3) A two-branch neural network is developed to determine the scheduling policy and the prediction horizons.

- We build a prototype system including a real-world robotic arm and its digital model in Metaverse, where the Nvidia Issac Gym platform is used. Extensive experiments are carried out in the prototype to evaluate the proposed task-oriented cross-system design approach. The experimental results show that our C-PPO outperforms several benchmark algorithms in terms of convergence time, stability, packet rate, and modeling error, and the cross-system design framework outperforms a baseline framework in terms of the required packet rate and the tail distribution of tracking errors.

The rest of this thesis is organized as follows. In Section 3.2, we propose the cross-system design framework and formulate a joint design problem that includes sampling, communications, and prediction components in one optimization problem. In Section 3.6, we develop the KC-TD3 algorithm to optimize the sampling and prediction policy while minimizing the communication load. Section 3.8 describes the prototype we used to verify our method and Section 3.9 provides performance evaluations. The Section 3.11 concludes the first technical work. For the second technical work, in Section 4.1, we propose the task-oriented cross-system design framework where all subsystems, i.e., sensing, communications, reconstruction, prediction, control, and rendering, are elaborated in detail. In Section 4.3, we develop the C-PPO algorithm to optimize the scheduling and prediction policy while minimizing the communication load under the constraint of CVaR. Section 4.5 describes the prototype and provides performance evaluations. Section 4.6 concludes the second technical work. Finally, chapter 5 concludes the thesis and discusses the open issues of task-oriented cross-system design for the Metaverse.

Chapter 3

Synchronizing the Real-World Device and Digital Model in Metaverse - Simple Case

3.1 Introduction

Facilitated by the rapid development of AR, VR and MR, the metaverse is expected to change our daily lives in different aspects, such as shopping, social interaction, healthcare [140], education [7] and gaming [141]. One of the main challenges is providing an immersive and highly interactive experience [142], which requires synchronization between the physical and virtual worlds to ensure smooth user motion tracking and timely feedback. Indeed, poor synchronization can lead to chaotic interactions and dizziness [143]. Besides, in mission-critical applications assisted by the Metaverse, even slight out-of-synchronization between a real-world device and the digital model may cause serious consequences [139].

The synchronization performance can be characterized by three key metrics: Motion-To-Photon (MTP) latency, data rate, and packet-loss rate. MTP latency refers to the time between a user's action and the corresponding effect displayed in the virtual world [144]. In applications that require haptic feedback, the required MTP should be less than 1 ms [145]. The data rate, i.e. the maximum rate of data transfer across the network, is the bottleneck for most multimedia applications [146], especially when a large amount of data is generated by multimedia sources – e.g. High-Definition video, AR/VR/MR, gaming, massive sensor networks, etc. Further, the packet-loss rate, i.e. the percentage of packets sent by a transmitter but not received by the receiver, is crucial for mission-critical applications, such as remote robotic control, smart-factories monitoring, and online healthcare [147].

It is very challenging to meet the three key performance metrics in practice. Although the three use cases have been considered the 5G cellular networks, eMBB, URLLC and mMTC, the requirements in metaverse cannot be fulfilled; indeed, while 5G New Radio would allow a sub-millisecond delay in the radio access network, the E2E delay is still far from the MTP

requirement [4]. Moreover, to support interactive applications globally, the Metaverse requires an extremely high data rate, far beyond the capabilities of 5G networks at 50 Gbits per second [9].

Recently, researchers started to investigate interdisciplinary approaches beyond conventional communication system designs. The existing literature has two branches of related work: sampling-communication co-design [97]–[102] and prediction-communication co-design [103]–[106]. (1) Sampling-communication co-design uses AoI, mutual information, or goal-oriented semantics to down-sample the information/packets at the transmitter side; thus, the task is completed with fewer communication load after reconstructing the original information at the receiver side. (2) Prediction-communication co-design, instead, predicts a device's future state based on its historical states; if the prediction horizon equals the communication delay, the user's experienced delay is zero [148]. In addition, if the communications present packet losses, the same approach can infer the missing states.

Considering that sampling, communications, and prediction are closely interconnected and interdependent, we argue that it is possible to improve the performance compared with the above existing work by combining the three. For example, suppose that a predictor can estimate missing information in a longer prediction horizon; in that case, the overall reliability is less sensitive to packet losses in communications, and the sampling rate can be reduced. Nevertheless, prediction and reconstruction errors may deteriorate the user experience; still, the impact on the synchronization in the Metaverse remains unclear, and a coherent design framework that combines sampling, communications, and prediction is not available in the existing literature.

In this work, we demonstrate how to synchronize a real-world robotic arm and its digital model in the Metaverse by sampling, communications, and prediction cross-system design. We establish a framework for minimizing the average communication load under the constraint of the average tracking error between a real-world robotic arm and its digital model. Then, we propose the KC-TD3 algorithm to adjust the sampling rate and the prediction horizon, where expert knowledge and advanced reinforcement learning techniques are exploited. In addition, we build a prototype of the proposed real-time robotic control system with a digital model in the Metaverse. The results of our experiments show that the proposed cross-system framework can significantly reduce the communication load in practical scenarios with communication packet losses. Compared to several benchmarks, our KC-TD3 algorithm converges faster and can guarantee the average tracking error constraint.

3.2 Cross-System Design Framework

In this section, we describe the proposed framework and formulate the cross-system design problem as the foundation of our algorithm. As shown in 3.1, we consider the synchronization between a real-world device and its digital model in the Metaverse. Specifically, a sensor of the device first measures its trajectory. Then, the data are sampled, i.e. decimated, and transmitted to

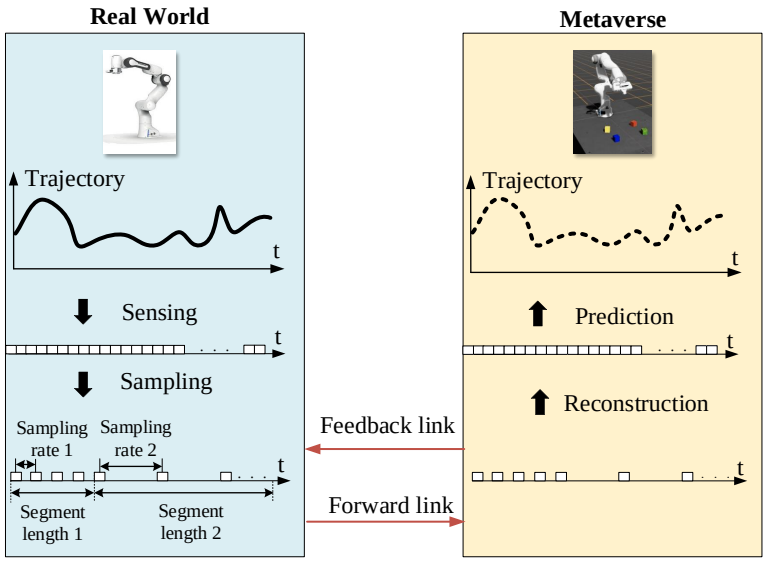


Figure 3.1: Proposed cross-system design framework to synchronize a real-world device and its digital model in the Metaverse.

the Metaverse, where it can be reconstructed and used to predict the future trajectory. To reduce the latency between the real-world devices and the digital model in the Metaverse, the digital model of the device follows the predicted trajectory and feeds back the prediction results to the real-world devices. Finally, the device compares its trajectory with the predicted one and adjusts the sampling rate and the prediction horizon. As shown in Fig. 3.2, time is discretized into slots. The E2E delays in the forward and feedback links of long-distance communication networks are denoted by D_d and D_f (slots), respectively. We assume that both delays are bounded by $D_{\max}$, i.e. if the delay of a packet is longer than $D_{\max}$, it is dropped by the system.

The trajectory of the device is measured by the sensor in each time slot. We can subdivide raw sensor measurements into sequences, which we call *segments*; we denote the k-th segment by $\mathscr{T}(k) = \{\tau_k(i) \mid i = 1, \dots W_k\}$, where $\tau_k(i)$ are sensor readings and W_k is the length of the segment $\mathscr{T}(k)$. We assume that the length of each segment is larger than the communication delay. As such, the k-th segment can arrive at the receiver by the end of the $(k+1)$-th segment. Thus, we have the following constraint

$$W_k \geq D_{\max}. \tag{3.1}$$

Since the raw trajectory samples are highly correlated in the time domain – and thus the data packets – they are highly redundant. For this reason, to reduce the communication load, we can subsample the raw data from the sensor and then transmit it to the cloud server via the long-distance communication network. Each trajectory segment is sampled at a fixed rate $r(k)$. Let $\tilde{\mathscr{T}}(k) = \{\tilde{\tau}_k(i) \mid i = 1, \dots n_k\}$ be the sampled trajectories of the k-th segment, where n_k is the number of packets sent by the transmitter.

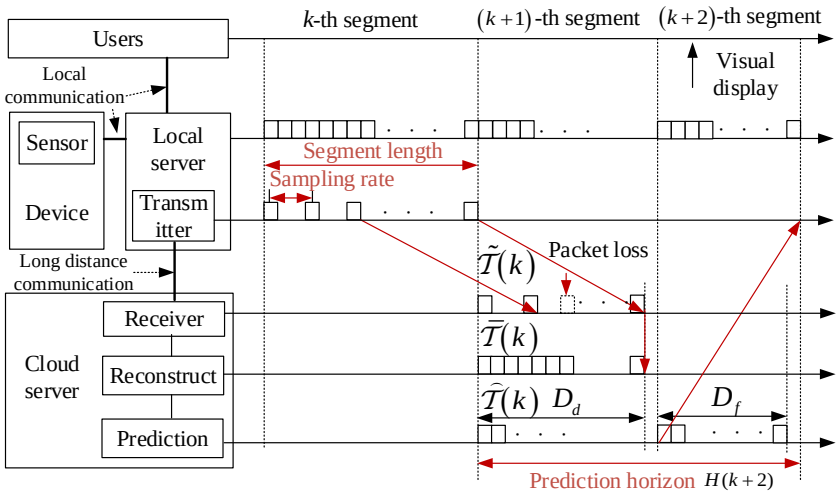


Figure 3.2: The timing sequence of the proposed cross-system design framework (The senor belongs to the real-world device for data generation and the transmitter belongs to a local server; The receiver and functions for construction and predication are deployed at the cloud server that operates the Metaverse).

After the reception, the cloud server reconstructs the original trajectory from the sampled one – we denote it by $\bar{\mathcal{T}}(k) = \{\bar{\tau}_k(i) \mid i = 1, \dots W_k\}$ – and, given the last historical trajectories $\bar{\mathcal{T}}(1), \bar{\mathcal{T}}(2), \dots \bar{\mathcal{T}}(k)$, predicts the future trajectory of the device, i.e. $\hat{\mathcal{T}}(k+1)$ and $\hat{\mathcal{T}}(k+2)$ of length W_{k+1} and W_{k+2}, respectively. As such, the prediction horizon needs to be

$$H(k+2) = W_{k+1} + W_{k+2}. \tag{3.2}$$

Considering that the communication delay in the feedback link does not exceed $D_{\max}$, which is shorter than W_{k+2}, the device obtains $\hat{\mathcal{T}}(k+2)$ by the end of the $(k+2)$-th segment, $\mathcal{T}(k+2)$. By measuring the average tracking error, which is defined as the MSE between the true trajectory, $\mathcal{T}(k+2)$, and the predicted trajectory, $\hat{\mathcal{T}}(k+2)$, the system can adjust the sampling rate and the length of the $(k+3)$-th segment W_{k+3}. According to (3.2), by adjusting the length of each segment, we adjust the prediction horizon. Besides, the sensing data of the real-world device are displayed to the user by different equipment (e.g., High-Definition Screen, AR, VR, glasses) via local communication networks (e.g., High-Definition Multimedia Interface (HDMI), or other outputs ports).

3.3 Sampling, Reconstruction, and Prediction

3.3.1 Sampling

The transmitter only sends the sampled packets to the receiver, residing on the remote server. Denoting the number of samples of the k-th segment by n_k, the sampling rate of the k-th segment is given by

$$r(k) = \frac{n_k}{W_k}. \tag{3.3}$$

As a result, the sampled trajectory can be obtained as follows,

$$\tilde{\tau}_k(i) = \tau_k\left(\left\lfloor \frac{W_k}{n_k} \right\rfloor \cdot (i-1) + 1\right), i = 1, ..., n_k. \tag{3.4}$$

where $\lfloor \cdot \rfloor$ represents the floor function.

3.3.2 Reconstruction

In digital signal processing, there are several up-sampling methods for state reconstruction, such as inverse fast Fourier transform, repetitions of the last received state, linear interpolation, and barycentric interpolation [149]. Without loss of generality, due to its wide adoption, and its simplicity, we adopt linear interpolation [150]. The relationship between the reconstructed trajectory and the sampled trajectory is given by

$$\bar{\mathscr{T}}(k) = f_{\mathrm{I}}(\tilde{\mathscr{T}}(k); \theta_{\mathrm{I}}), \tag{3.5}$$

where θ_{I} are the parameters for the interpolation function.

3.3.3 Prediction

We utilize a MLP that takes the historical trajectory as its input and generates the predicted one. The relationship between the input and output of the predictor in the t_{k+1}-th slot can be expressed as

$$\begin{aligned}
&[\hat{\tau}_{k+1}(t_{k+1} - W_{k+1} + 1), ..., \hat{\tau}_{k+1}(t_{k+1} + W_{k+2})] \\
&= f_{\mathrm{P}}(\bar{\tau}(t_{k+1} - L_{\mathrm{in}} - W_{k+1}), \bar{\tau}(t_{k+1} - L_{\mathrm{in}} - W_{k+1} + 1), \\
&\quad ..., \bar{\tau}(t_{k+1} - W_{k+1}); \theta_{\mathrm{P}}),
\end{aligned} \tag{3.6}$$

where L_{in} is the length of the historical trajectory and θ_{P} represents the MLP parameters. We ignore the subscript k of the $\bar{\tau}$ since the input of the predictor includes multiple trajectory

segments. It is worth noting that the value of L_{in} is determined by the temporal correlation of the trajectory and does not depend on the length of the k-th segment, W_k.

3.4 Tracking Error and Communication Load

3.4.1 Tracking Error

The tracking error of the k-th segment, denoted by $e(k)$, is given by

$$\begin{aligned} e(k) &= \text{MSE}(\mathscr{T}(k), \hat{\mathscr{T}}(k)) \\ &= \frac{1}{|W_k|} \sum_{t=1}^{W_k} (\tau_k(i) - \hat{\tau}_k(i))^2, \end{aligned} \tag{3.7}$$

where $\tau_k(i)$ is the trajectory measured at the i-th time slot in the k-th segment, and $\hat{\tau}_k(i)$, for $i = 1, ..., W_k$, is the prediction trajectory.

3.4.2 Communication Load

We assume that the communication load is proportional to the packet rate of the system. For example, in Orthogonal Frequency-Division Multiplexing (OFDM) communication systems (adopted in the 4-th generation and 5-th generation cellular networks), the time-frequency resource blocks occupied by a packet are determined by the packet size and the channel gain [9]. If the packet size and channel gain are stationary, the average resource blocks allocated to a device are proportional to the packet rate (i.e., the sampling rate in our system). Since the number of packets to be transmitted for the k-th segment is n_k, the required radio resource blocks to transmit the k-th segment of the trajectory are proportional to n_k. Therefore, to minimize the required radio resource blocks for this service, we minimize the sampling rate subject to a constraint on the MSE in (3.7).

3.5 Problem Formulation

To reduce the communication load subject to the MSE constraint, we optimize the sampling rate and the prediction horizon. The problem can be formulated as follows,

$$\min_{H(k), n_k} \lim_{N \to +\infty} \frac{1}{N} \sum_{k=1}^{N} r(k) \tag{3.8}$$

$$\text{s.t.} \quad \lim_{N \to +\infty} \frac{1}{N} \sum_{k=0}^{N} e(k) \leq \Gamma_c, \tag{3.8a}$$

where N represents the number of segments and Γ_c is the required average-MSE threshold; this latter is dependent on specific applications and is considered fixed for both training and deployment.

3.6 Constrained Deep Reinforcement Learning for Optimizing Sampling, Prediction, and Communications

The problem (3.8) is a sequential decision problem and can be re-formulated as a Constrained Markov Decision Process (CMDP). To solve the problem, we integrate expert knowledge into the primal-dual Deep Deterministic Policy Gradient (DDPG) algorithm and develop the KC-TD3 algorithm.

3.6.1 CMDP Formulation

State: The state observed by the device by the end of the k-th segment (the t_k-th time slot) includes the last two trajectory segments measured by the device, $\mathscr{T}(k-1)$ and $\mathscr{T}(k)$, and those predicted by the cloud server, $\hat{\mathscr{T}}(k-1)$ and $\hat{\mathscr{T}}(k)$. The predicted trajectory segments depend on the reconstructed historical trajectory $\bar{\mathscr{T}}_{in}(k) = \bar{\tau}(t_k - L_{\text{in}} - W_k - W_{k-1}), ..., \bar{\tau}(t_k - W_k - W_{k-1})$. As shown in (3.4) and (3.5), $\bar{\mathscr{T}}_{in}(k)$ is determined by the true trajectory from the $(t_k - L_{\text{in}} - W_k - W_{k-1})$-th slot to the $(t_k - W_k - W_{k-1})$-th slot, as well as the sampling rate and the prediction horizon.

Action: The action to be taken by the end of the k-th segment includes the length of the $(k+1)$-th segment, W_{k+1}, and the number of samples to be transmitted n_{k+1} (equivalent to the sampling rate (r_{k+1})). Thus, the action is denoted by $a_k = (W_{k+1}, n_{k+1})$. For convenience, we denote two actions as a pair denoted by $a_k = [a_k^1, a_k^2]$, where $a_k^1 = W_{k+1}$ and $a_k^2 = n_{k+1}$. Based on this definition, we have $a_k^1 \in \{D_{\max}, ..., W_{\max}\}$ and $a_k^2 = 1, ..., a_k^1$.

Instantaneous Reward and Cost: Given the state and action at the end of the k-th segment, the instantaneous reward is the negative of the sampling rate and the cost is the MSE of the $(k+1)$-th segment. According to (3.3) and (3.7), we have $r_k = -r(k+1)$ and $c_k = e(k+1)$.

Policy: The agent follows a deterministic policy denoted by $\mu : a_k = \mu(s_k|\theta)$, where θ represents the parameters of the policy.

Long-Term Reward and Long-Term Cost: Following the policy $\mu(\cdot|\theta)$, the long-term discounted reward is given by

$$R^{\mu(\cdot|\theta)} = \mathbb{E}[\textstyle\sum_{k=0}^{\infty} \gamma^k r_k], \tag{3.9}$$

where the γ is the discount factor. Similarly, the long-term discounted cost under the policy $\mu(\cdot|\theta)$ is given by

$$C^{\mu(\cdot|\theta)} = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k c_k]. \tag{3.10}$$

CMDP Formulation: The goal is to find the optimal policy $\mu^*(\cdot|\theta^*)$ that maximizes the long-term reward $R^{\mu(\cdot|\theta)}$ subject to the constraint on the long-term cost $C^{\mu(\cdot|\theta)}$. Thus, the problem can be reformulated as follows:

$$\mu^*(\cdot|\theta^*) = \arg\max_{\mu(\cdot|\theta)} R^{\mu(\cdot|\theta)} \tag{3.11}$$

$$\text{s.t.} \quad C^{\mu(\cdot|\theta)} \leqslant \frac{\Gamma_c}{1-\gamma}. \tag{3.11a}$$

To utilize DRL to solve the problem, we first prove that the transitions of the system follow an MDP (see the proof in Appendix A). The Lagrangian function of the constrained optimization problem is defined as follows [151],

$$\Gamma(\mu(\cdot|\theta), \lambda) = R^{\mu(\cdot|\theta)} - \lambda(C^{\mu(\cdot|\theta)} - \frac{\Gamma_c}{1-\gamma}), \tag{3.12}$$

where λ is the Lagrangian multiplier. Then, the constrained problem can be converted to the following unconstrained problem,

$$(\mu^*(\cdot|\theta^*), \lambda^*) = \arg\min_{\lambda \geqslant 0} \max_{\mu(\cdot|\theta)} \Gamma(\mu(\cdot|\theta), \lambda). \tag{3.13}$$

3.6.2 Preliminary of Primal-Dual DDPG

Primal-dual DDPG is an off-policy method to solve Constrained Markov Decision Processes (CMDP) [152]. It combines the DDPG algorithm with the primal-dual method to find the optimal policy and the dual variable. The policy, the long-term reward, and the long-term cost are represented by three neural networks, and we denote them by $\mu(\cdot|\theta)$, $Q^R(\cdot|\phi^R)$ and $Q^C(\cdot|\phi^C)$, where θ, ϕ^R, and ϕ^C are the parameters of these three neural networks, respectively.

During the training process, the corresponding action generated by the actor-network is given by

$$\mathbf{a}_k = clip\left(\mu(\mathbf{s}_k|\theta) + clip(\varepsilon, -c, c), \mathbf{a}_{\text{Low}}, \mathbf{a}_{\text{High}}\right), \tag{3.14}$$

where ε is the White Gaussian Noise with distribution $\varepsilon \sim \mathcal{N}(0, \sigma)$, $[\mathbf{a}_{\text{Low}}, \mathbf{a}_{\text{High}}]$ is the action space, and

$$clip(x, c_1, c_2) = \min(\max(x, c_1), c_2). \tag{3.15}$$

After taking action $\mathbf{a}_k$ at the k-th step (i.e., selecting W_k and n_k in the last time slot of the k-th trajectory segment in our system), the system observes the instantaneous reward and cost, and transits from $\mathbf{s}_k$ to $\mathbf{s}_{k+1}$. The transition is denoted by $\mathscr{D}_k \triangleq \langle \mathbf{s}_k, \mathbf{a}_k, r_k, c_k, \mathbf{s}_{k+1} \rangle$, which is stored in the replay memory, $\mathscr{M}$. In each training step, a number of transitions (mini-batch) are randomly selected from replay memory $\mathscr{M}$ and are used to optimize θ, ϕ^R, and ϕ^C. We denote $\mathscr{D}_{k_i} = \langle \mathbf{s}_{k_i}, \mathbf{a}_{k_i}, r_{k_i}, c_{k_i}, \mathbf{s}_{k_i+1} \rangle, i = 1, 2, ..., N_{\text{batch}}$ as the i-th transition in the k-th episode of the training stage, where N_{batch} is the batch size. To optimize the critic and cost neural networks, the Bellman equation is utilized. The target reward and cost functions can be expressed as follows:

$$Q^R(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}) = r + \gamma Q^R\left(\mathbf{s}_{k_i+1}, \mu(\mathbf{s}_{k_i+1}|\theta)|\phi^R\right), \tag{3.16}$$

$$Q^C(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}) = r + \gamma Q^C\left(\mathbf{s}_{k_i+1}, \mu(\mathbf{s}_{k_i+1}|\theta)|\phi^C\right). \tag{3.17}$$

Then, the long-term reward and long-term reward cost neural networks are updated by using the mean-squared Bellman error (MSBE) loss function which is derived as follows

$$L\left(\phi^R\right) = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \left[\left(Q^R(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}) - Q^R\left(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}|\phi^R\right)\right)^2\right], \tag{3.18}$$

$$L\left(\phi^C\right) = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \left[\left(Q^C(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}) - Q^C\left(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}|\phi^C\right)\right)^2\right]. \tag{3.19}$$

In primal-dual DDPG, the actor policy is updated by maximizing the Lagrangian function in (3.12), where the long-term reward and the long-term cost are replaced by the critic network and the cost network, respectively, i.e.,

$$\max_{\theta} \mathbb{E}\left[Q^R(\mathbf{s}_{k_i}, \mathbf{a}_{k_i}) - \lambda Q^C(\mathbf{s}_{k_i}, \mathbf{a}_{k_i})\right]. \tag{3.20}$$

After that, the dual variable λ is updated by gradient descent to minimize the Lagrangian function according to

$$\lambda^{(k+1)} = \left[\lambda_i^{(k)} + \beta_k\left(Q^C\left(\mathbf{s}_{k_i}, \mu(\mathbf{s}_{k_i}|\theta)|\phi^C\right) - \frac{\Gamma_c}{1-\gamma}\right)\right]^+, \tag{3.21}$$

where β_k is the step size and $[x]^+ = \max\{0, x\}$.

3.7 KC-TD3 Design

The straightforward application of primal-dual DDPG in our problem can cause several issues (to be discussed in the following). To improve the performance in the training process of primal-dual DDPG, we propose to exploit advanced reinforcement learning techniques and expert knowledge on sampling, communications, and prediction including 1) extension of double Q-Learning, 2) state-space reduction, 3) interdependent action normalization, and 4) APDO. The resulting approach is KC-TD3.

3.7.1 Extension of Double Q-Learning

The overestimation bias of the critic network will lead to poor performance when optimizing the actor-network [23]. Similarly, with primal-dual DDPG, if the cost network is underestimated, the constraint cannot be satisfied. To address the first issue, two critic networks are trained to approximate the state-action value function in double Q-Learning, where the target value of the Bellman equation is the smaller one of the two critic networks. Thus, the target state-action reward value $Q^R(s_{k_i}, a_{k_i})$ is expressed by

$$Q^R(s_{k_i}, a_{k_i}) = r + \gamma \min_{i=1,2} Q^R_{\phi_i^R}\left(s_{k_i+1}, \mu(s_{k_i+1}|\theta)|\phi_i^R\right), \tag{3.22}$$

where $Q^R_{\phi_i^R}$ is the estimated state-action value function from the i-th critic network with parameter ϕ_i^R. As an extension of double Q-Learning, the target cost values are also estimated by two cost networks, which can be expressed as follows:

$$Q^C(s_{k_i}, a_{k_i}) = r + \gamma \max_{i=1,2} Q^C_{\phi_i^C}\left(s_{k_i+1}, \mu(s_{k_i+1}|\theta)|\phi_i^C\right), \tag{3.23}$$

where $Q^C_{\phi_i^C}$ is the estimated state-action cost function of the i-th cost network with parameter ϕ_i^C.

3.7.2 State-Space Reduction

The original state $s(k)$ consists of two trajectory segments measured by the device, $\mathscr{T}_{in}(k-1)$ and $\mathscr{T}_{in}(k)$, and that predicted by the cloud server, $\hat{\mathscr{T}}_{in}(k-1)$ and $\hat{\mathscr{T}}_{in}(k)$. However, the dimension of the state is dynamic since the length of each trajectory segment depends on the action W_k. One possible approach is to replace the trajectory segments with the input of the prediction algorithm, $\bar{\mathscr{T}}_{\text{in}}$, which relies on the measured trajectory segments and determines the predicted trajectory segments. Although the dimension of $\bar{\mathscr{T}}_{\text{in}}$ is fixed, it can be large when the correlation time of the trajectory is large (up to a few seconds) and the state generation rate is high (1000 samples/s in our prototype). This may lead to a long training time and require a large number of samples. To

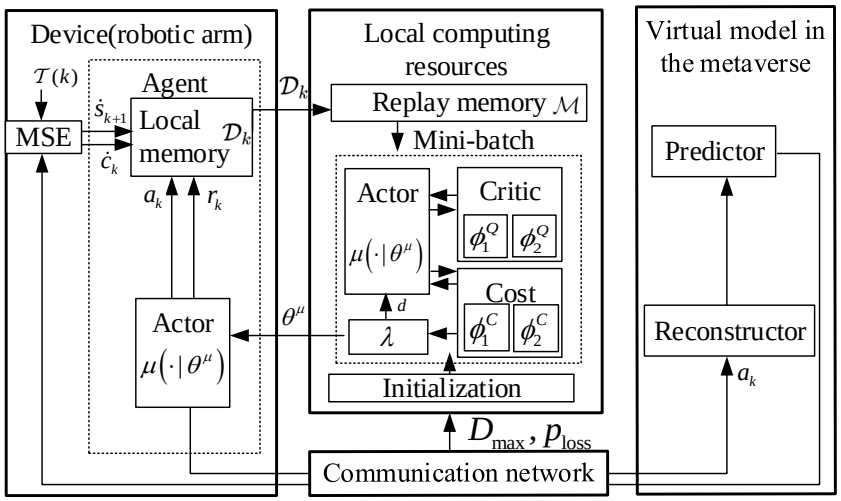


Figure 3.3: Illustration of proposed KC-TD3 architecture.

improve the learning efficiency, we replace the original state with the MSE in (6),

$$\dot{s}_k = e(k). \tag{3.24}$$

In this way, we can reduce the input of the actor-network to a scalar.

3.7.3 Interdependent Action Normalization

Considering the aforementioned action in Section 3.6.1, we design the action with two elements, $\left(a_k^1, a_k^2\right)$, where $a_k^1 \in \{D_{\max}, ..., W_{\max}\}$ and $a_k^2 = 1, ..., a_k^1$. The feasible region of the second element depends on the first element, we normalize a_k^2 by using W_k (inverse normalization by a_k^1), i.e.,

$$\dot{a}_k^2 = \frac{a_k^2}{W_k}. \tag{3.25}$$

With this normalization, we have $\dot{a}_k^2 \in [0, 1]$. Thus, we can use a sigmoid function in the output layer of the actor-network.

3.7.4 APDO

The dual variable updating procedure only utilizes on-policy samples, which leads to low sampling efficiency. To address this issue, we proposed to apply APDO [152], where the dual variable is updated by the dual gradient ascent every d_λ iteration. With this approach, historical data samples stored in the replay buffer are utilized to update λ and help improve sample efficiency and convergent speed.

3.7.5 KC-TD3 Training Architecture

The proposed KC-TD3 structure can be implemented in the real-world system with the architecture in Fig. 3.3, which mainly consists of a device (such as a robotic arm), local computing resource (a desktop/local server connected to the robotic arm), the communication network, and the Metaverse.

3.7.6 Communication Network Initialization

The delay, $D_{\max}$, and the packet loss probability, p_{loss}, in the communication network are measured in the initialization stage. With linear reconstruction in (3.5), we need at least two samples to reconstruct each trajectory segment. Otherwise, the system is in the outage. We denote the outage probability by ζ. For a given requirement on the outage probability, such as 10^{-5}, and a packet loss probability in the communication network (up to 10 % in our experiments), the minimum number of samples the transmitter should update for each segment is denoted by $N_{\min}$. To meet the outage probability requirement, $N_{\min}$ can be obtained from the following expression,

$$p_{\text{loss}}^{N_{\min}} + N_{\min} p_{\text{loss}}^{N_{\min}-1}(1-p_{\text{loss}}) \leq \zeta. \tag{3.26}$$

3.7.7 Training Algorithm

The details of the algorithm can be found in Algorithm 1. With the observed average tracking error, the agent at the device takes actions according to the output of the actor network. Then, the average tracking error of the next trajectory segment is measured and saved in the replay memory. After that, the transitions in the replay memory are randomly selected to optimize the critic and cost networks (referred to as one iteration of the gradient descent optimization). Finally, the actor-network and the dual variable λ are updated every d_{a} and d_{λ} iterations, respectively.

3.8 Prototype Design and Data Collection

3.8.1 Prototype Design

We build a prototype[1] as shown in Fig. 3.4, where a virtual robotic arm needs to synchronize with a physical robotic arm in the real world. This is essential for many future use cases, such as education, healthcare, Industry 4.0, etc.

Physical Robotic Arm in Real World: An industrial-grade robotic arm system, Franka Emika Panda [153], is used in our prototype. It has seven Degrees of Freedom (DoF) with the

[1] In our prototype, we simplify the system by considering the one-way synchronization from a single physical robotic arm to a virtual robotic arm, and test our cross-system design framework and KC-TD3 algorithm. The proposed cross-system design can be extended to multi-user cases.

Algorithm 1 KC-TD3

Input: Initialize parameters of actor, critic and cost networks, θ, ϕ_1^R, ϕ_2^R, ϕ_1^C, ϕ_2^C. Measure the latency and packet loss probability in the communication network, $D_{\max}$ and p_{loss}. Obtain the minimal number of samples the device needs to update for each trajectory segment from (3.26).

Output: Optimal $\lambda^*$and optimal policy $\mu^*(\cdot|\theta^*)$.

Initialize the target networks: $\theta_{\text{arg}} \leftarrow \theta$, $\phi^R_{\text{arg},1} \leftarrow \phi_1^R$, $\phi^R_{\text{arg},2} \leftarrow \phi_2^R$, $\phi^C_{\text{arg},1} \leftarrow \phi_2^C$, $\phi^C_{\text{arg},2} \leftarrow \phi_2^C$.
Initialize the Lagrangian multiplier $\lambda = 0$, actor update delay d_{a}, dual variable update delay d_λ.
for episode m = 1,... **do**
  Observe the average tracking error $\dot{s}_k$.
  Generate an action based on (3.14) and execute the action.
  Observe the reward, cost, the next state, and store $\langle \dot{s}_{k_i}, \mathbf{a}_{k_i}, r_{k_i}, c_{k_i}, \dot{s}_{k_i+1} \rangle$ in memory $\mathscr{M}$.
  for j in range (from 1 to the maximal number of updates) **do**
    Randomly sample a batch of transitions $\langle \dot{s}_{k_i}, \mathbf{a}_{k_i}, r_{k_i}, c_{k_i}, \dot{s}_{k_i+1} \rangle$ from $\mathscr{M}$.
    Updating Q^R-functions by one step of gradient descent based on (3.16), (3.18).
    Updating Q^C-functions by one step of gradient descent based on (3.17), (3.19).
    if j mod $d_{\text{a}} = 0$ **then**
      Update the actor policy by one step of gradient ascent based on (3.20).
    end if
    if k mod $d_\lambda = 0$ **then**
      Update dual variable by one step of gradient ascent based on (3.21).
    end if
    Update target network by:

$$\begin{aligned} \phi^R_{\text{arg}} &\leftarrow \rho \phi^R_{\text{arg},i} + (1-\rho)\phi_i^R; \\ \phi^C_{\text{arg}} &\leftarrow \rho \phi^C_{\text{arg},i} + (1-\rho)\phi_i^C; \\ \theta_{\text{arg}} &\leftarrow \rho \theta_{\text{arg}} + (1-\rho)\theta; \\ i &= \{1,2\}. \end{aligned}$$

  end for
end for

speed limit of the end-effector at 2 m/s. The robotic arm receives the target end-effector position from a controller and then conducts inverse-kinematics calculations to map the target end-effector position to the seven joint angle positions of the robotic arm. After that, the robotic arm applies a propotional-integral-differential (PID) controller [154], which converts the joint angle positions to a series of commands on the angular velocity of each joint. At the same time, the robotic system outputs the actual angle position of each joint every 1 ms, which is the data used in our KC-TD3 algorithm.

Virtual Robotic Arm in the Metaverse: Unity software is used to generate the virtual robotic arm in the Metaverse [155]. Specifically, we construct the digital model of the physical robotic arm, Franka Emika Panda, with the same number of DoFs deployed on a cloud server. The virtual robotic arm needs to obtain the angle position of each joint in real-time. Such that the virtual

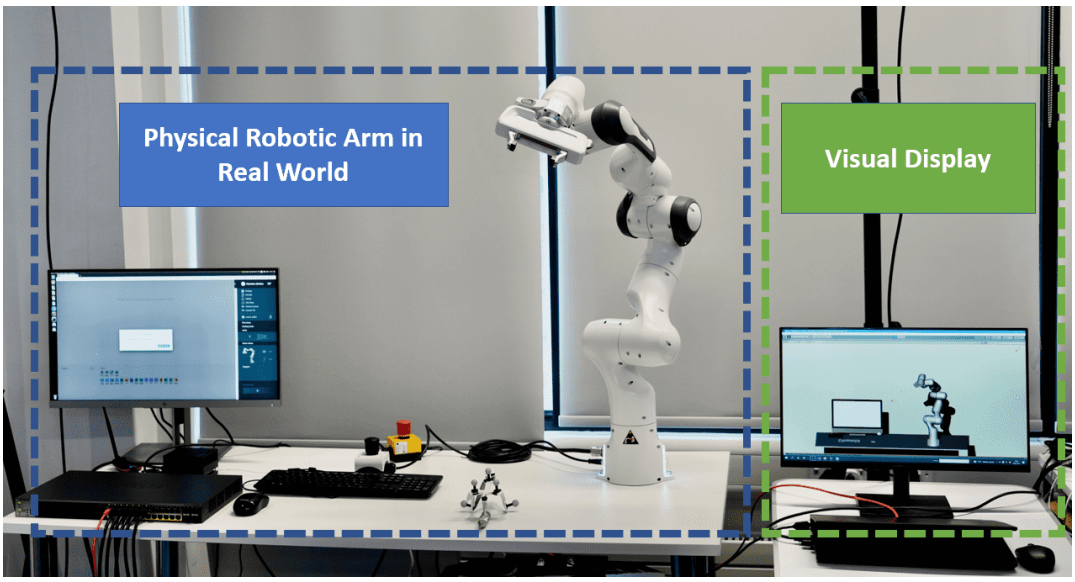


Figure 3.4: Our prototype system (the movements of a physical robotic arm and the visual display. The digital model in the Metaverse to be synchronized is in the remote which is not shown in the graph).

Parameters	Values
Slot duration	1 ms
Transmission time interval T	1 ms
Time delay bound $D_{\max}$	10 ms
The range of segment length $\mathscr{T}(k)$	10 ms to 100 ms
The range of sampling rate $n(k)$	20 Hz to 1000 Hz
Experimental time	2×10^4 ms

Table 3.1: System Parameters for Performance Evaluation

robotic arm can be synchronized with the physical robotic arm. In the communication system, our prototype uses User Datagram Protocol (UDP) to connect the virtual robotic arm to the physical one. Besides, the sensing data of the robotic arm are displayed to the user by High-Definition Screen via HDMI.

3.8.2 Data Collection

As shown in Fig. 3.5, a human operator controls the physical robotic arm via a motion capture system [156], drawing a "star" shape in the air for 20 seconds. Specifically, six OptiTrack Prime-13 motion capture cameras are deployed in a 4×4 m^2 area, where the human operator holds a tool with seven passive markers [156]. The motion capture system constructs the seven markers as a rigid body and outputs the position (3 DoFs) and orientation (3 DoFs) of the rigid body at the

Figure 3.5: Illustration of our data collection via an experiment, where a human operator controls the physical robotic arm to draw the "star" shape in the air (The demonstration video of our data collection is available at ^_ https : //youtu.be/LCqSGtkrgug)

.

frequency of 120 Hz. The robotic arm receives the position and orientation as the target position of its end-effector. By controlling the angles of the seven joints, the physical robotic arm is able to move its end-effector toward the target position. In this way, the end-effector of the robotic arm tracks the human operator's hand trajectory in real-time. In this work, we use the data of the first joint from the base to demonstrate our design, where the system parameters are provided in Table 3.1.

3.8.3 Operations Complexity Discussion

Overall, the complexity of the experimental manipulation was not high. With the designed algorithm, the user's movements were smoothed to a certain extent, which could allow the user to adapt to the controls and complete the movements in a short period of time. In addition, we used the same real-world control platform in [157]. In the experiment, 30 participants for the interactive control experiments, all of whom were using the platform for the first time. Approval from the Institutional Review Board (IRB) has been granted under IRB #4392. Prior to the experiments, participants are given a 5-minute window to acquaint themselves with the platform through the operation of the robotic arm. The results show that the operators can successfully perform a series of tasks including writing the letters "S", "W", "Z" and "ABC", drawing the curves including "Circle", "Five-pointed Star" and "Triangle" in the air and perform the basic

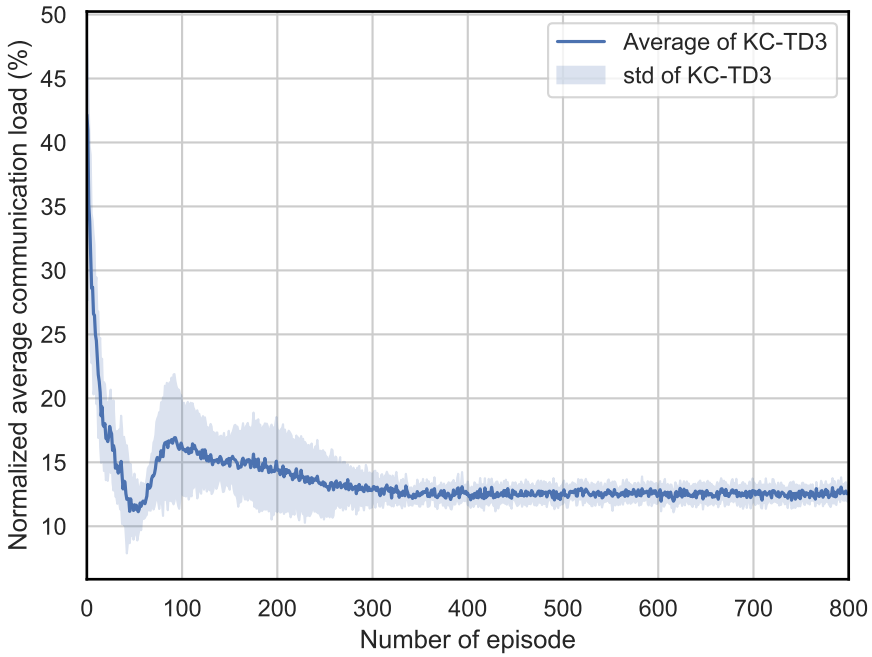


Figure 3.6: Normalized average communication load in each episode.

operations such as “Pointing”, “Pushing” and “Stirring”. The results illustrate that our operation complexity is acceptable for a wide range of operators.

3.9 Performance Evaluation

In this section, we first evaluate the training performance of our KC-TD3 and then compare the performance of the proposed sampling, communications, and prediction cross-system design framework with different benchmarks.

3.9.1 KC-TD3 Training Performance

The average KC-TD3 training performance and standard deviation are illustrated in Figs. 3.6-3.8. In each training experiment, we provide the results in 800 episodes. Then, we repeat the same experiment 10 times. In Fig. 3.6, we provide the normalized average communication load, where the communication load without sampling is 100 %. In Figs. 3.7 and 3.8, the average tracking error and the value of the dual variable are presented. The results in Figs. 3.6-3.8, show that the KC-TD3 algorithm converges to a stable policy after 300 episodes. The normalized average communication load is 13%, and the average tracking error is MSE $= 0.007^{\circ}$, which satisfies the average tracking error constraint.

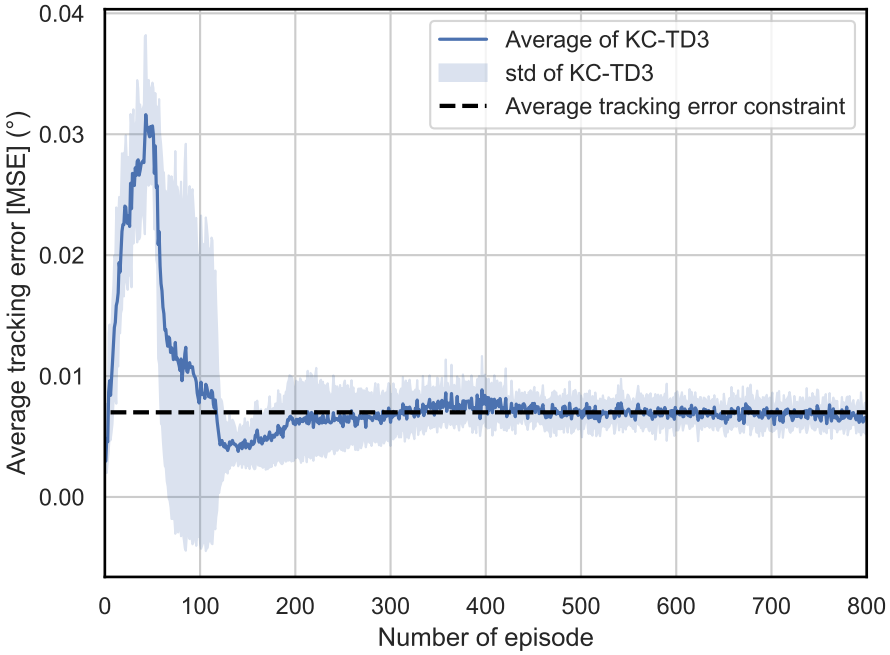


Figure 3.7: Average tracking error in each episode.

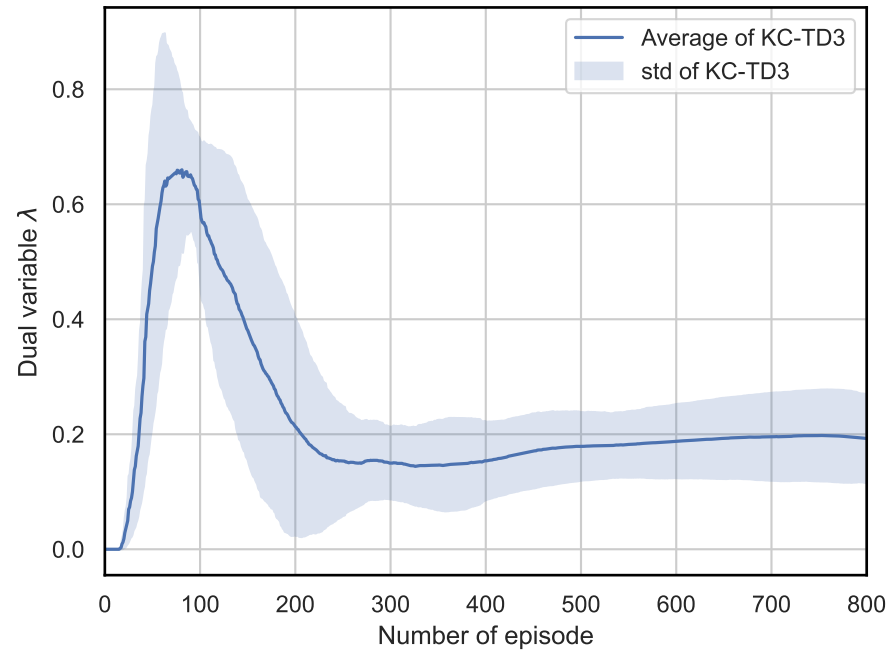


Figure 3.8: Dual variable in each episode.

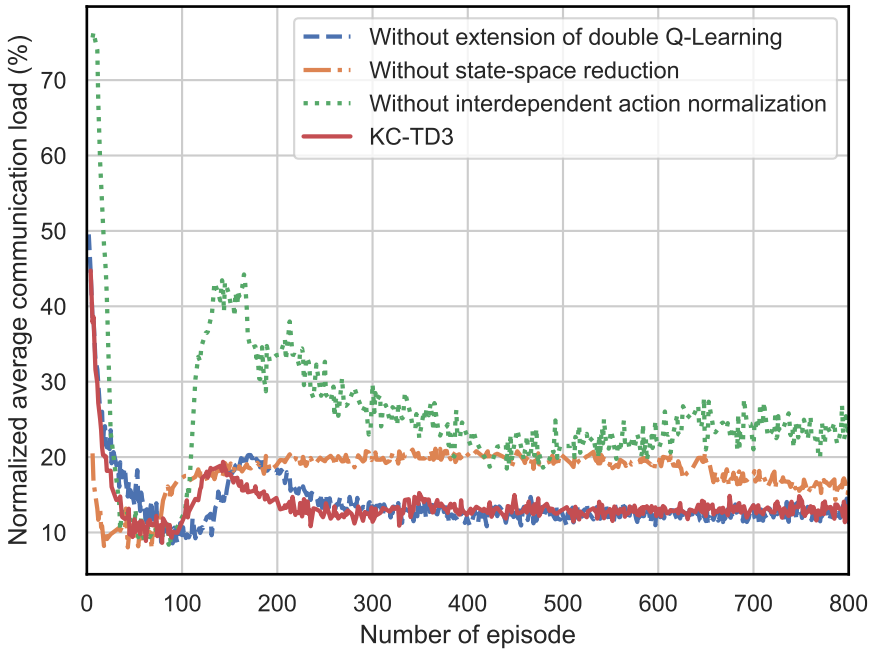


Figure 3.9: Normalized average communication load in each episode.

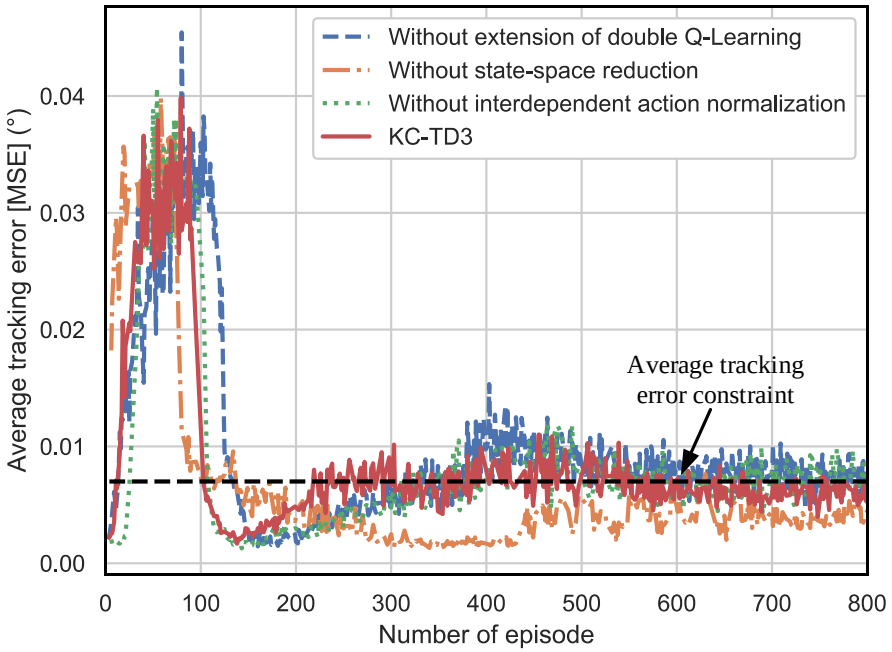


Figure 3.10: Average tracking error in each episode.

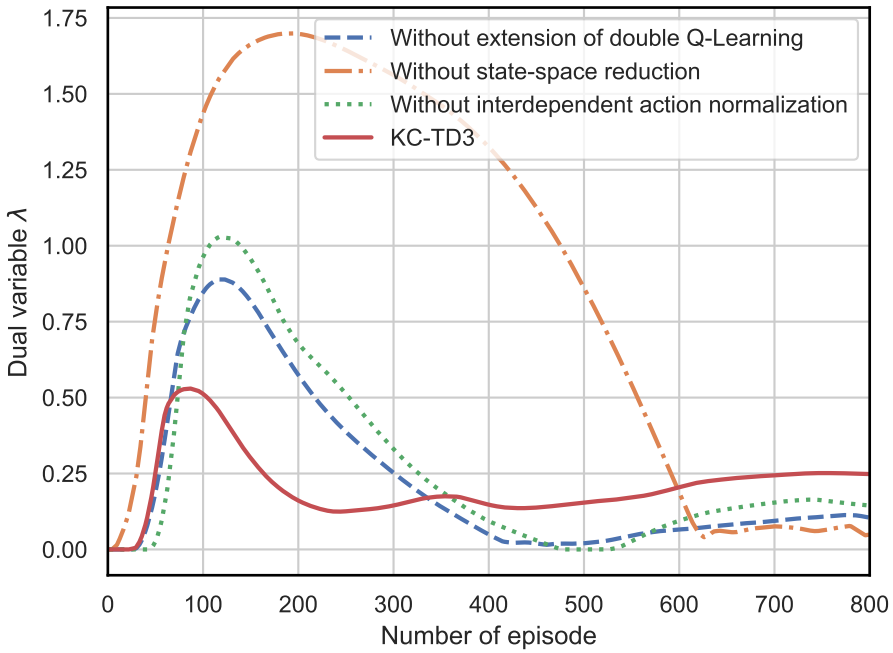


Figure 3.11: Dual variable in each episode.

3.9.2 Ablation Study on Expert Knowledge

In Figs. 3.9-3.11, we illustrate the impacts of different expert knowledge on the training performance of the proposed KC-TD3. When only partial expert knowledge is available, we obtained three benchmarks: (a) KC-TD3 without extension of double Q-Learning; (b) KC-TD3 without state-space reduction; (c) KC-TD3 without interdependent action normalization. From Figs. 3.9-3.11, we observe that the proposed strategy with full expert knowledge has the best performance in terms of stability and convergence. It also achieves the lowest communication load (in Fig. 3.9), and satisfies the average tracking error constraint (in Fig. 3.10). By comparing the proposed KC-TD3 algorithm with the three benchmarks, we can obtain the following insights: (1) without the extended double Q-Learning, the algorithm can hardly meet the average tracking error constraint; (2) without state-space reduction, the algorithm cannot converge to the optimal solution, where the average tracking error is far below the constraint at the cost of a higher communication load. (3) Without the interdependent action normalization, the obtained policy can meet the average tracking error constraint, but it does not minimize the communication load.

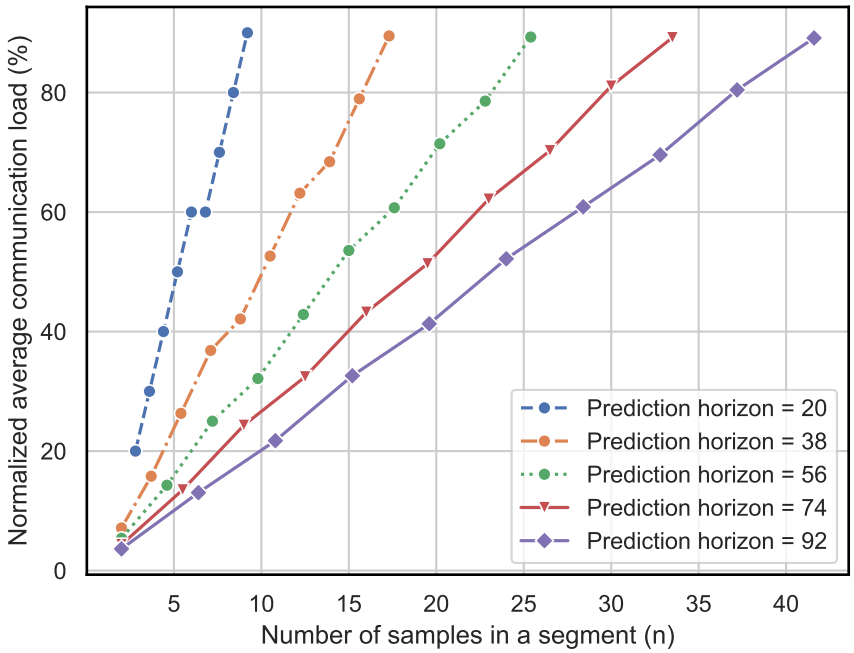


Figure 3.12: Normalized average communication load under different prediction horizons and sampling numbers.

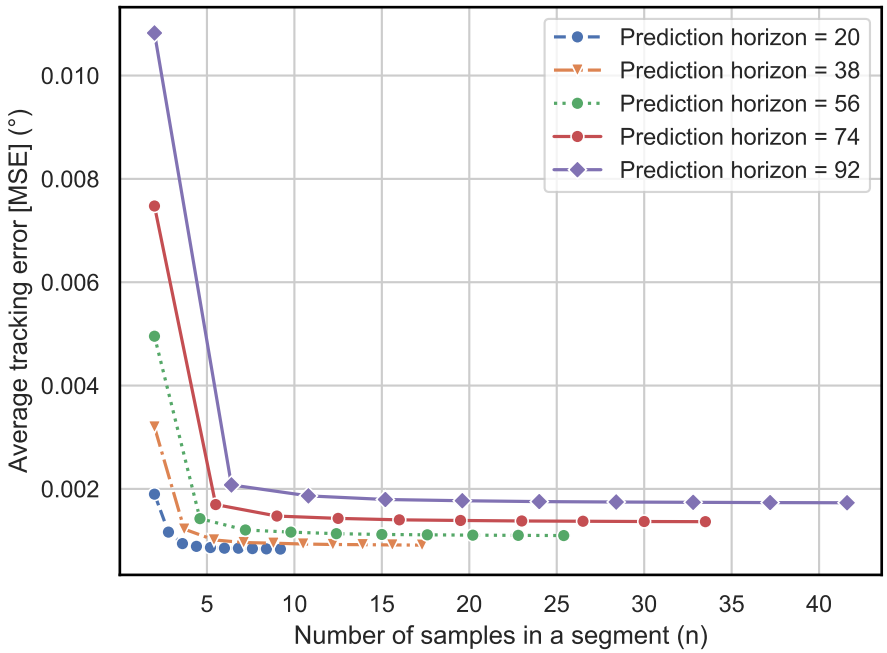


Figure 3.13: Average tracking error under different prediction horizons and the number of samples.

	Baseline	Exhaustive search	KC-TD3
Average tracking error constraint = 0.002°			
Normalized average communication load (%)	100%	27%	27%
Average tracking error (°)	0.016	0.0020	0.0019
Average tracking error constraint = 0.007°			
Normalized average communication load (%)	100%	13%	13%
Average tracking error (°)	0.016	0.0070	0.0069

Table 3.2: Performance Comparisons of Different Design

3.10 Validation of the Sampling, Communications, and Prediction Cross-System Design Framework

Fixed Sampling Policy

We first evaluate the performance of a benchmark policy with a fixed sample rate and prediction horizon. Specifically, Fig. 3.12 shows the normalized average communication load versus the number of samples in each segment. The results show that the normalized average communication load grows as the number of samples in each segment increases, but decreases as the prediction horizon increases. This means that a higher sampling rate (the ratio of the number of samples in each segment to the duration of the segment, which is half of the prediction horizon) leads to a higher communication load.

We further provide the average tracking errors with different sampling rates in Fig. 3.13. The results show that the average tracking error decreases dramatically as the number of samples in each segment n increases from 2 to 10. When $n > 10$, the average tracking error is nearly constant. In addition, a longer prediction horizon leads to a higher average tracking error. Therefore, the number of samples in each segment and the segment length should be optimized to obtain the minimum normalized average communication load subject to the average tracking error constraint.

3.10.1 Overall Performance

Fig. 3.14 compares the tracking errors of different design approaches, where the delay bound is $D_{\max} = 50$ ms and the average tracking error constraint is MSE = 0.007°. The "Baseline"

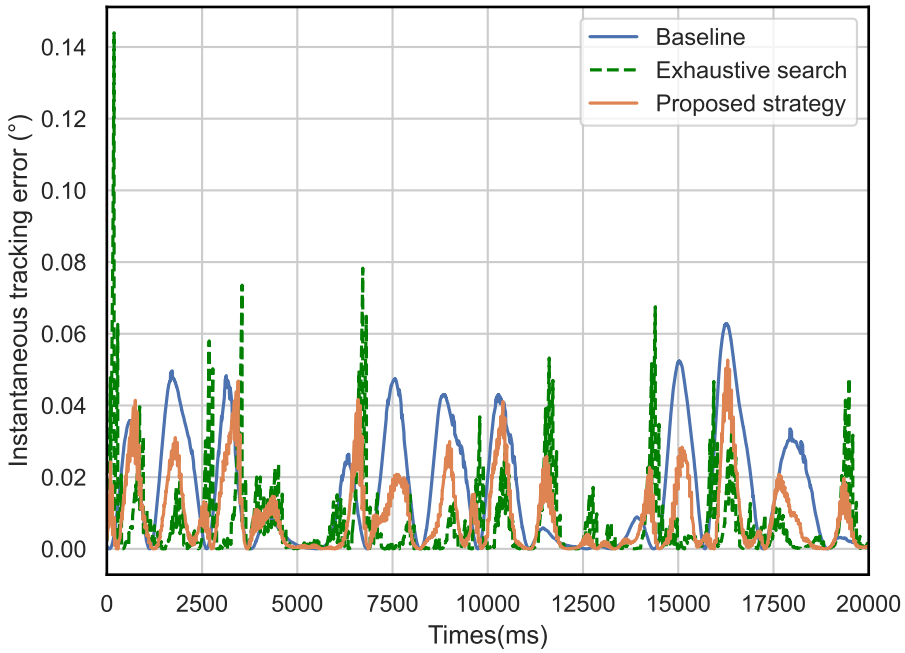


Figure 3.14: Instantaneous tracking error of Baseline, exhaustive search, and proposed KC-TD3, where the E2E latency is 50 ms and the average MSE constraint is 0.007°.

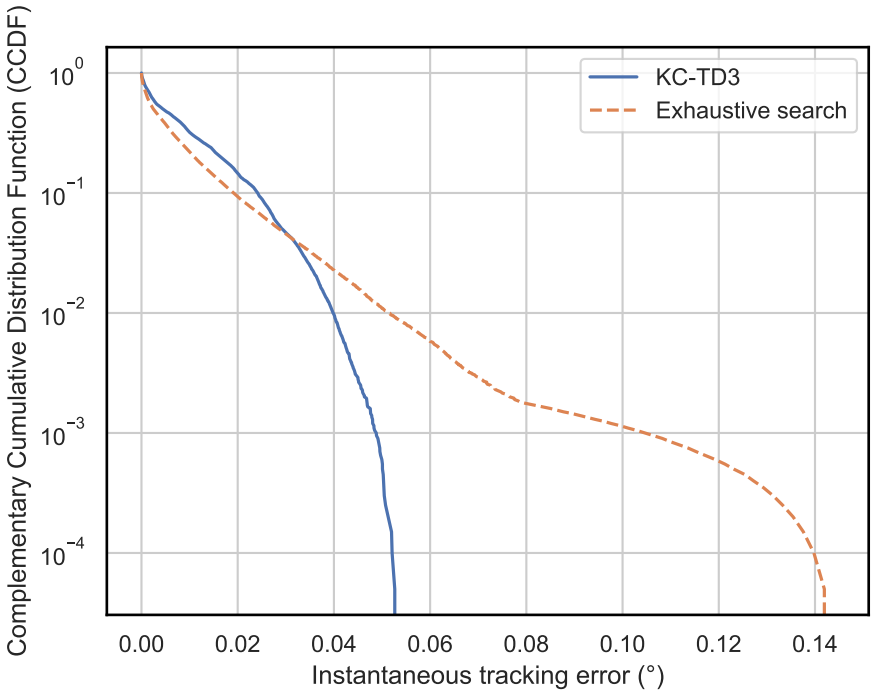


Figure 3.15: CCDF comparison of exhaustive search and proposed KC-TD3, where the E2E latency is 50 ms and the average MSE constraint is 0.007°.

approach transmits all samples to the receiver with the communication load of 100% and there is no prediction at the receiver side. The "Exhaustive search" approach is obtained by searching a fixed sampling rate and a fixed prediction horizon that minimizes the normalized average communication load subject to the constraint on the average tracking error. We propose it here only as a performance benchmark since it is not feasible for a practical online application. The instantaneous tracking errors achieved by the proposed strategy and the above two approaches are provided in Fig. 3.14, which shows that the proposed KC-TD3 can significantly reduce the instantaneous tracking error. This means that the synchronization between the virtual robotic arm and the physical one can be effectively improved.

Table 3.2 compares KC-TD3 with "Baseline" and "Exhaustive search" approaches. Compared with the "Baseline" approach, KC-TD3 reduces the normalized average communication load by 73% and improves the average tracking error by 87.5% (when the average tracking error constraint is 0.002°). When the average tracking error constraint is 0.007°, KC-TD3 reduces the normalized average communication load by 87% and improves the average tracking error by 56.2% compared with the "Baseline" approach. Besides, the normalized average communication load and the average tracking error achieved by KC-TD3 are the same as the "Exhaustive search" approach.

Furthermore, we compared our policy that dynamically adjusts the sampling rate and prediction horizon according to the MSE with the exhaustive search approach that optimizes a static sampling rate and a static prediction horizon. The exhaustive search approach represents an optimal method that cannot be achieved online due to resource constraints. In Fig. 3.15, we provide the comparison results of the CCDF of the tracking error. The results indicate that the CCDF achieved by "Exhaustive search" has a longer tail compared with the CCDF achieved by KC-TD3. This means that by adjusting the sampling rate and the prediction horizon dynamically, KC-TD3 can effectively reduce the tail probability of the tracking error.

Fig. 3.16 further demonstrates the trade-off between the normalized average communication load and the average tracking error achieved by KC-TD3, where different packet loss probabilities in the communication system are considered, i.e., $p_{\text{loss}} = 0, 1\%,$ and 10%. The results reveal a trade-off between the normalized average communication load and the average tracking error. In addition, with a smaller packet loss probability, it is possible to achieve a better trade-off between the normalized average communication load and the average tracking error. Furthermore, compared with the "Baseline" approach, KC-TD3 can reduce up to 75% of normalized average communication load subject to a 0.002° average tracking error constraint even when the packet loss probability in the communication system is as high as 10%.

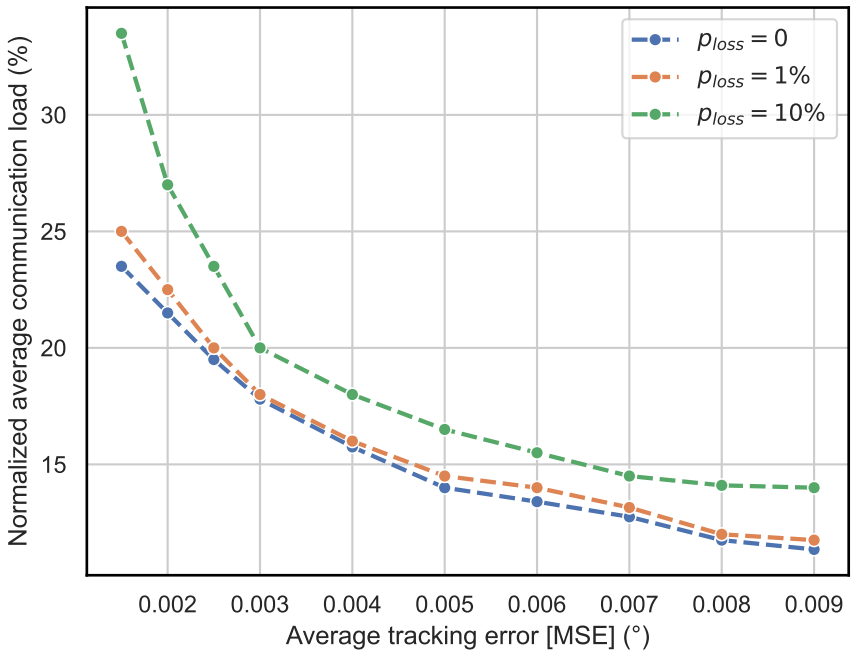


Figure 3.16: Trade-off between normalized average communication load and average tracking error with different packet loss probabilities $p_{\text{loss}} = 0$, 1% and 10%.

3.11 Conclusions

In this work, we demonstrated how to synchronize a real-world robotic arm and its digital model in the metaverse by sampling, communications, and prediction co-design. We established a framework for minimizing the average communication load under the constraint of the average tracking error between a real-world robotic arm and its digital model. Then, we proposed the KC-TD3 algorithm to adjust the sampling rate and the prediction horizon, where expert knowledge and advanced reinforcement learning techniques are exploited. In addition, we built a prototype of the proposed real-time robotic control system with a digital model in the metaverse. The results of our experiments showed that the proposed cross-system design framework can significantly reduce the communication load in practical scenarios with communication packet losses. When the tracking error constraint is stringent (MSE $= 0.002^\circ$), our policy degenerates into the policy in the existing sampling-communication co-design framework. When the tracking error constraint is mild (MSE $= 0.007^\circ$), our policy degenerates into the policy in the existing prediction-communication co-design framework. Furthermore, compared to several benchmarks, our KC-TD3 algorithm achieves better convergence time, stability, communication load, and average tracking error.

Chapter 4

Synchronizing the Real-World Device and Digital Model in Metaverse - General Case

As a digital world that mirrors the physical world and generates feedback for human users in real-time, the Metaverse can blur the lines between the physical and digital worlds and revolutionize how humans communicate and interact with each other [4]. To achieve this goal, timely and accurate modeling of real-world devices/humans in the Metaverse is critical for user experience. The communications and computing latency and digital model distortion will lead to chaotic interactions and user dizziness [143]. For some mission-critical applications assisted by Metaverse, such as remote healthcare and factory automation, slight out-of-synchronization between a real-world device and its digital model may cause serious consequences [139].

While 5G networks have significantly improved latency, reliability, data rate, and connection density, they still fall short of satisfying the demands of the Metaverse [9]. One of the examples is that 5G New Radio can achieve 1 ms latency and 10^{-5} packet error probability in the RANs, but does not guarantee end-to-end delay and reliability. Jitter is another issue that can lead to inaccurate modeling, which may be caused by 5G RANs and 5G core networks. Furthermore, there is a mismatch between communication KPIs, i.e., latency, reliability, jitter, and throughput, and the KPIs requirements of diverse tasks in the Metaverse, such as modeling error, haptic feedback distortion, and semantic segmentation errors, which will lead to poor user experience and low resource utilization efficiency.

On the other hand, the performance of modeling in the Metaverse is not solely determined by communication networks. Other systems, including sensing, prediction, control, and rendering, can also have significant impacts on E2E latency and accuracy. Designing these systems separately results in strictly sub-optimal solutions and may fail to meet the task-oriented KPI requirements. Thus, cross-system design has been investigated in the recent literature, such as prediction and communication co-design [10], [106] and sampling and communication co-design [95], [158]. They have shown significant gains in the cross-system design, but they have also revealed potential issues. For example, cross-system models could be analytically intractable, and the complexity of

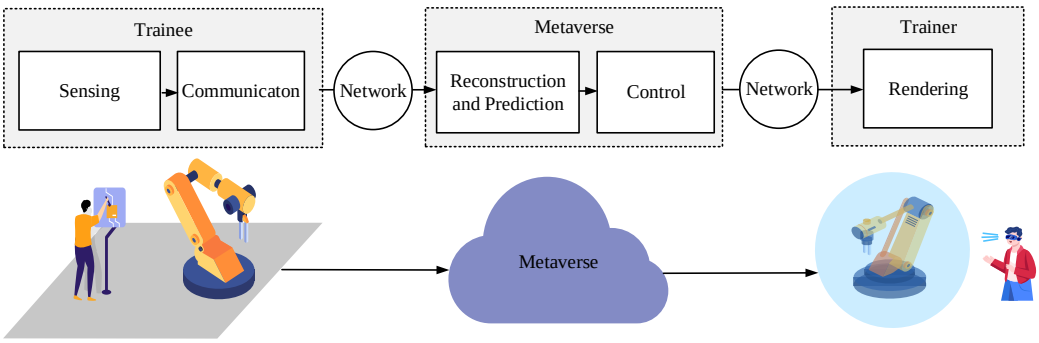


Figure 4.1: Proposed task-oriented cross-system design framework for modeling a robotic arm in the Metaverse, where sensing, communications, reconstruction, predication, control, and rendering are considered.

cross-system problems can be much higher than problems in separate design approaches. Thus, novel methodologies are needed for cross-system design.

The first step toward cross-system design is to formulate a problem that takes the relevant systems into account. Nevertheless, it could be difficult to obtain closed-form expressions of the objective function and the constraints as some of the KPIs are analytically intractable. Although we can use some approximations to formulate the problem, it is generally non-convex or NP-hard. This motivates us to develop data-driven deep learning approaches, where the policy to be optimized is represented by a neural network. DRL is a promising method for training the neural network. For example, PPO is developed to optimize policies with a discrete action space [159]. More recently, the primal-dual method and Constraint-Rectified Policy Optimization (CRPO) were introduced into DRL for solving constrained problems [152], [160]. It is worth noting that a straightforward implementation of DRL algorithms may not work [161]. Integrating domain knowledge from relevant systems into DRL algorithms is essential for the success of DRL [79] in practical applications.

4.1 Task-Oriented Cross-System Design Framework

In this section, we propose a task-oriented cross-system design framework for timely and accurate modeling in the Metaverse. The specific goal is to build a digital model of a real-world robotic arm for real-time monitoring and control.

The framework is shown in Fig. 4.1, where the real-world robotic arm with multiple joints is controlled by a user (a student or a trainee) for some tasks for example training in healthcare. The trajectory (the angles of all the joints) of the real-world robotic arm is measured by the built-in sensors. Then, the trajectory is sampled and transmitted to the Metaverse in a cloud server, where the sampled data are used to reconstruct the historical trajectory and predict the future trajectory. Here, the digital model in the Metaverse is controlled by the predicted trajectory rather than the

reconstructed trajectory to compensate for delays caused by different components across systems. Finally, the digital model in the Metaverse is rendered and presented to another user (an expert or a trainer) via a computer screen or AR/VR headset. It is worth noting that each joint has its own state, and the states of the joints are interdependent. They need to collaborate with each other to accomplish the task. In addition, the total communication resources shared by all the joints are limited. Thus, it is possible to extend our system into multi-sensor scenarios.

The E2E MTP latency is defined by the delay between the movement of the real-world robotic arm and the movement of its digital model in the Metaverse. Thus, it includes communication delay, computation delay, control delay, and rendering delay. By optimizing the prediction horizon and the scheduling policy[1], we minimize communication overhead subject to constraints on the modeling accuracy and the MTP latency.

Fig. 4.2 illustrates the timing sequence of the proposed framework. The data is generated from the built-in sensors at the physical robotic arm. Then, the communication module conducts scheduling and sends the selected data to a computer server via a network. The server conducts data reconstruction and prediction and then controls the digital model of the robotic arm. Finally, the digital model was rendered[2] and presented to a human (trainer) via a VR headset (or a screen). In the following, we will introduce each component:

4.1.1 Sensing and Communications

Time is discretized into slots with a duration of t_s. The built-in sensors measure I joint angles in each time slot. Let $\mathscr{T}(t) = [\tau_1(t), ..., \tau_I(t)]$ be the trajectory of the real-world robotic arm, where $\tau_i(t)$, $i = 1, 2, 3, ..., I$, is the angle value of the i-th joint measured in the t-th time slot. We consider a scheduling policy in the communication system, where the indicator, $x_i(t)$, represents whether the i-th joint is scheduled for data transmission in the t-th time slot, $i = 1, ..., I$. If the i-th joint is not scheduled, then $x_i(t) = 0$. Otherwise, $x_i(t) = 1$, and one packet will be transmitted to the Metaverse. The decision of the scheduler in the t-th time slot is denoted by $X(t) = [x_1(t), x_2(t), ..., x_I(t)]$. The total number of packets to be transmitted in communication systems in the t-th time slot is given by $\sum_{i=1}^{I} x_i(t)$.

4.1.2 Reconstruction

To reconstruct the trajectory from sampled data, we use the linear interpolation and extrapolation method, which is widely used in the existing literature and can be easily implemented in our system [150]. The indicators of packet arrivals in the t-th time slot at the Metaverse are denoted by $y_i(t)$, $i = 1, 2, ..., I$. If a packet from the i-th joint arrived at the Metaverse in the t-th time slot, then $y_i(t) = 1$. Otherwise, $y_i(t) = 0$. From the arrived packets, the set of joint angles obtained by

[1] The scheduling policy determines which joints will be scheduled for data transmission.

[2] To simplify the system, we assume that the rendering takes place at the server and the human user (trainer) directly interacts with the digital model via HCIs.

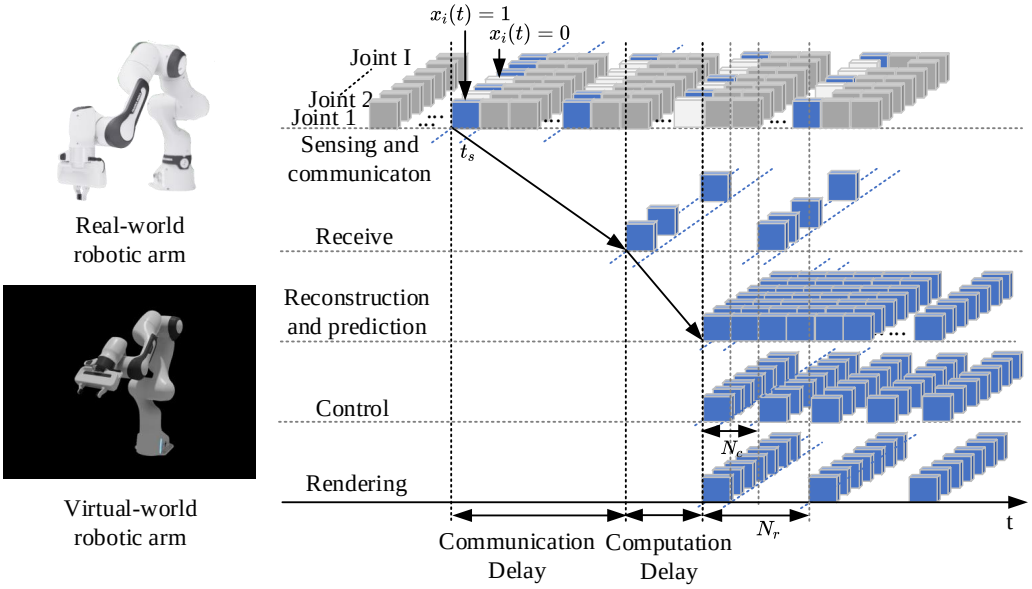


Figure 4.2: The timing sequence of the proposed framework, where the modeling accuracy and the MTP latency need to be satisfied.

the Metaverse in the t-th time slot is denoted by $\mathbb{S}(t) = \{\tau_i(t) \mid y_i(t) = 1,\ i = 1,...,I\}$. In a certain time slot t_0, the cloud server reconstructs the trajectory of the robotic arm from the received joint angles in a historical observation window with a duration of W_l time slots. The reconstruction algorithm is given by

$$\bar{\mathcal{T}}(t_0) = F_l(\mathbb{S}(t), \theta_l \mid t \in [t_0 - W_l, t_0 - W_l + 1, ..., t_0 - 1]), \tag{4.1}$$

where $\bar{\mathcal{T}}(t_0) \in \mathbb{R}^{1\times I}$ is the reconstructed trajectory, $F_l(\cdot, \theta_l)$ is the reconstruction function, and θ_l is the interpolation and extrapolation parameters.

4.1.3 Prediction

To compensate for the MTP latency, we propose to use the attention-mechanism-based predictor, referred to as *Informer*, to predict the future trajectory from the historical trajectory [162]. The lengths of the input and output trajectories are denoted by W_p and H. The values of W_p and H are determined by the auto-correlation coefficient of the trajectories [163]. We denote the prediction results for trajectory in the t-th time slot by $\hat{\mathcal{T}}(t) = [\hat{\tau}_1(t), \hat{\tau}_2(t), ..., \hat{\tau}_I(t)]$. In a certain time slot t_1, the prediction algorithm can be expressed as follows,

$$[\hat{\mathcal{T}}(t_1+1), \hat{\mathcal{T}}(t_1+2), ..., \hat{\mathcal{T}}(t_1+H)] = F_p([\bar{\mathcal{T}}(t_1 - W_p), \bar{\mathcal{T}}(t_1 - W_p + 1), ..., \bar{\mathcal{T}}(t_1)], \theta_p), \tag{4.2}$$

where $F_p(\cdot, \theta_p)$ is the prediction function and θ_p represents the parameters of this function. The loss function of the prediction algorithm is MSE between the output trajectory and the ground truth, which is given by

$$L_{\mathrm{p}} = \frac{1}{H} \sum_{n=1}^{H} \left(\hat{\mathscr{T}}(t_1+n) - \mathscr{T}(t_1+n) \right)^2. \tag{4.3}$$

We will optimize the prediction length $Z(t) = [z_1(t), z_2(t), ..., z_I(t)], z_i(t) \leq H$ for each joint in our cross-system design framework, which will be introduced in the next section.

4.1.4 Control

There are different algorithms we can use to control the virtual robotic arm in Metaverse [164]. Without loss of generality, we utilize the joint-space position control and Proportional–Derivative (PD) controller [165]. Considering the limitations of the control frequency and the processing time, the target angle for each joint will be generated by the control algorithm and subsequently executed for every N_{c} time slot, which is denoted by $\widetilde{\mathscr{T}}(t) = [\widetilde{\tau}_1(t), ..., \widetilde{\tau}_I(t)]$. In the t_2-th time slot, for each joint i, the target joint position $\widetilde{\tau}_i(t_2)$ to be executed within the N_{c} time slots can be expressed as

$$\widetilde{\tau}_i(t_2) = k_p \cdot (\hat{\tau}_i(t_1 + z_i(t)) - \widetilde{\tau}_i(t_2 - N_c)) + k_d \cdot (\frac{\mathrm{d}\hat{\tau}_i(t_1 + z_i(t))}{\mathrm{d}t} - \frac{\mathrm{d}\widetilde{\tau}_i(t_2 - N_c)}{\mathrm{d}t}), \tag{4.4}$$

where k_p and k_d are the proportional and derivative parameters of the PD controller, respectively.

4.1.5 Rendering

In computer graphics, rendering refers to the process of generating controllable and photo-realistic images and videos of virtual scenes [166]. In our system, the processing time of each image is denoted by N_r time slots. In other words, the monitor or AR/VR glasses refresh the images at a refresh rate of $1/(N_r t_{\mathrm{s}})$ (times/second). The relationship between the trajectory of the digital model and the trajectory displayed to the user is given by

$$\hat{\mathscr{T}}(t) = F_r(\widetilde{\mathscr{T}}(t), \theta_r), \tag{4.5}$$

where $F_r(\cdot, \theta_r)$ is the rendering function and θ_r represents the parameters for rendering.

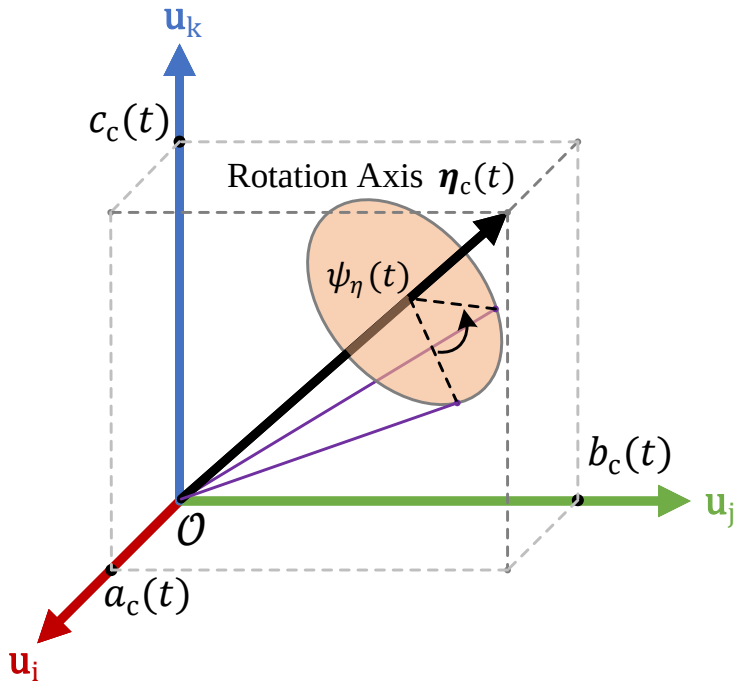


Figure 4.3: Orientation of the end effector to the base coordinate system $\mathscr{O}$.

4.2 KPIs and Communication Load

4.2.1 Task-Oriented KPI

The end effector of a robotic arm could be a gripper, a drill bit, or a sensor, depending on the specific task. We assume that the real-world end effector has seven degrees of freedom, and the pose of the end effector is denoted by

$$\mathscr{P}(t) = [l_{\mathrm{x,r}}(t), l_{\mathrm{y,r}}(t), l_{\mathrm{z,r}}(t), q_{\mathrm{x,r}}(t), q_{\mathrm{y,r}}(t), q_{\mathrm{z,r}}(t), q_{\mathrm{w,r}}(t)]. \tag{4.6}$$

Specifically, $l_{\mathrm{x,r}}(t)$, $l_{\mathrm{y,r}}(t)$, $l_{\mathrm{z,r}}(t)$ are the coordinates of the end effector in a three-dimensional Cartesian coordinate system. $[q_{\mathrm{x,r}}(t), q_{\mathrm{y,r}}(t), q_{\mathrm{z,r}}(t), q_{\mathrm{w,r}}(t)]$ is the quaternion representing the orientation of the end effector [167]. Quaternions are preferred over other representations, such as Euler angles or rotation matrices, in our context because of their compact representation and their ability to avoid a particular limitation associated with 3D rotation systems, known as gimbal lock, which can cause a loss of degrees of freedom [168]. Please see Appendix C for more information.

As shown in Fig. 4.3, the unit vector of the rotation axis, $\boldsymbol{\eta}_c(t) = [a_c(t), b_c(t), c_c(t)]$, and angle of rotation, $\psi_\eta(t)$, can be characterized by $q_{\mathrm{x,r}}(t)$, $q_{\mathrm{y,r}}(t)$, $q_{\mathrm{z,r}}(t)$, and $q_{\mathrm{w,r}}(t)$. In particular, the rotation axis is located in the coordinate system defined by three imaginary unit basic vectors, $\mathbf{u}_i$, $\mathbf{u}_j$, and $\mathbf{u}_k$, which follow special multiplication rules [167]. The relationship among the

quaternions, $\boldsymbol{\eta}_{\mathrm{c}}(t)$, and $\psi_{\eta}(t)$ is expressed by

$$\begin{aligned} q_{\mathrm{x,r}}(t) &= \sin(\frac{\psi_{\eta}(t)}{2}) \cdot a_{\mathrm{c}}(t), \\ q_{\mathrm{y,r}}(t) &= \sin(\frac{\psi_{\eta}(t)}{2}) \cdot b_{\mathrm{c}}(t), \\ q_{\mathrm{z,r}}(t) &= \sin(\frac{\psi_{\eta}(t)}{2}) \cdot c_{\mathrm{c}}(t), \\ q_{\mathrm{w,r}}(t) &= \cos(\frac{\psi_{\eta}(t)}{2}). \end{aligned} \tag{4.7}$$

Similarly, the quaternions of the virtual-world robotic arm follow the same rules.

From the I joint angles, $\mathscr{P}(t)$ is obtained from the forward kinematics according to

$$\mathscr{P}(t) = F_f(\mathscr{T}(t)), \tag{4.8}$$

where $F_f(\cdot)$ maps the joint angles to the pose of the end effector (positions and orientations). The expression of (4.8) depends on the structure and configuration of the robotic arm. Like $\mathscr{P}(t)$, the pose of the end effector displayed to the user also has seven degrees of freedom, denoted by $\check{P}(t) = [l_{\mathrm{x,v}}(t), l_{\mathrm{y,v}}(t), l_{\mathrm{z,v}}(t), q_{\mathrm{x,v}}(t), q_{\mathrm{y,v}}(t), q_{\mathrm{z,v}}(t), q_{\mathrm{w,v}}(t)]$. A task-oriented KPI is defined as the Euclidean distance between $\mathscr{P}(t)$ and $\check{P}(t)$,

$$\begin{aligned} \mathbf{e}(t) = {} & \omega_1 \cdot \|(l_{\mathrm{x,r}}(t), l_{\mathrm{y,r}}(t), l_{\mathrm{z,r}}(t)), (l_{\mathrm{x,v}}(t), l_{\mathrm{y,v}}(t), l_{\mathrm{z,v}}(t))\|_2 \\ & + \omega_2 \cdot \|(q_{\mathrm{x,r}}(t), q_{\mathrm{y,r}}(t), q_{\mathrm{z,r}}(t), q_{\mathrm{w,r}}(t)), (q_{\mathrm{x,v}}(t), q_{\mathrm{y,v}}(t), q_{\mathrm{z,v}}(t), q_{\mathrm{w,v}}(t))\|_2, \end{aligned} \tag{4.9}$$

where $\|\cdot\|_2$ is the L_2-norm defined as $\|\cdot\|_2 \triangleq \sqrt{\sum(\cdot)^2}$, and ω_1 and ω_2 are the weighting coefficients. The first term is the position error and the second is the orientation error. Depending on the accuracy requirements of different tasks in the Metaverse, ω_1 and ω_2 can be set to different values.

4.2.2 Communication Load

The Orthogonal Frequency-Division Multiplexing (OFDM) communication system is considered in our framework for it is widely deployed in cellular networks. The number of time and frequency resource blocks allocated to a packet is determined by the channel gain and the packet size. We assume that the channel gain and the packet size are two stationary random variables. The average number of resource blocks required in each slot is proportional to the average packet rate. To improve resource utilization efficiency, defined as the average number of resource blocks per slot, we minimize the average number of packets transmitted in each slot.

4.3 Problem Fomulation

In this section, we formulate an optimization problem that minimizes the communication load subject to a constraint on the CVaR of modeling error by optimizing the scheduling policy and the prediction horizon. In the cross-system design framework, there is no closed-form relationship between the KPIs and the optimization variables. To solve this problem, we develop a DRL algorithm by integrating domain knowledge into the advanced PPO algorithm.

4.3.1 Preliminary of PPO

PPO is an advanced reinforcement learning algorithm for solving problems with discrete action space. We chose PPO as the baseline algorithm due to its simplicity, effectiveness, and high sample efficiency compared to other reinforcement learning algorithms [169]. In addition, PPO maintains a balance between exploration and exploitation and avoids drastic policy updates, which is crucial for managing the complex dynamics of robotics and ensuring stable training [170]. We denote the state and action of PPO by $\mathbf{s}_t$ and $\mathbf{a}_t$, respectively. The policy is a mapping from the state to the probabilities of taking different actions, denoted by $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$, where θ are the training parameters of the policy network. With PPO, the parameters of the policy are updated according to the following expression,

$$\theta_{t+1} = \arg\max_{\theta} \mathop{\mathbb{E}}_{\mathbf{s}_t,\mathbf{a}_t\sim\pi_\theta} \mathscr{L}(\mathbf{s}_t,\mathbf{a}_t,\theta_t,\theta). \tag{4.10}$$

The loss function $\mathscr{L}(\mathbf{s}_t,\mathbf{a}_t,\theta_t,\theta)$ is given by

$$\begin{aligned}\mathscr{L}(\mathbf{s}_t,\mathbf{a}_t,\theta_t,\theta) = \min\Bigg(&\frac{\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\theta_t}(\mathbf{a}_t \mid \mathbf{s}_t)} A^{\pi_\theta}(\mathbf{s}_t,\mathbf{a}_t), \\ &\operatorname{clip}\left(\frac{\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\theta_t}(\mathbf{a}_t \mid \mathbf{s}_t)}, 1-\varepsilon, 1+\varepsilon, \right) A^{\pi_{\theta_t}}(\mathbf{s}_t,\mathbf{a}_t)\Bigg),\end{aligned} \tag{4.11}$$

where $A(\mathbf{s}_t,\mathbf{a}_t)$ is the advantage function defined as the difference between the state-action value function, $Q^{\pi_\theta}(\mathbf{s}_t,\mathbf{a}_t)$, and the state value function, $V^{\pi_\theta}(\mathbf{s}_t)$,

$$A^{\pi_\theta}(\mathbf{s}_t,\mathbf{a}_t) = Q^{\pi_\theta}(\mathbf{s}_t,\mathbf{a}_t) - V^{\pi_\theta}(\mathbf{s}_t), \tag{4.12}$$

which estimates the advantage of taking action $\mathbf{a}_t$ in state $\mathbf{s}_t$, over other possible actions in the same state [171].

In the sequel, we develop our DRL algorithm by integrating domain knowledge into the PPO.

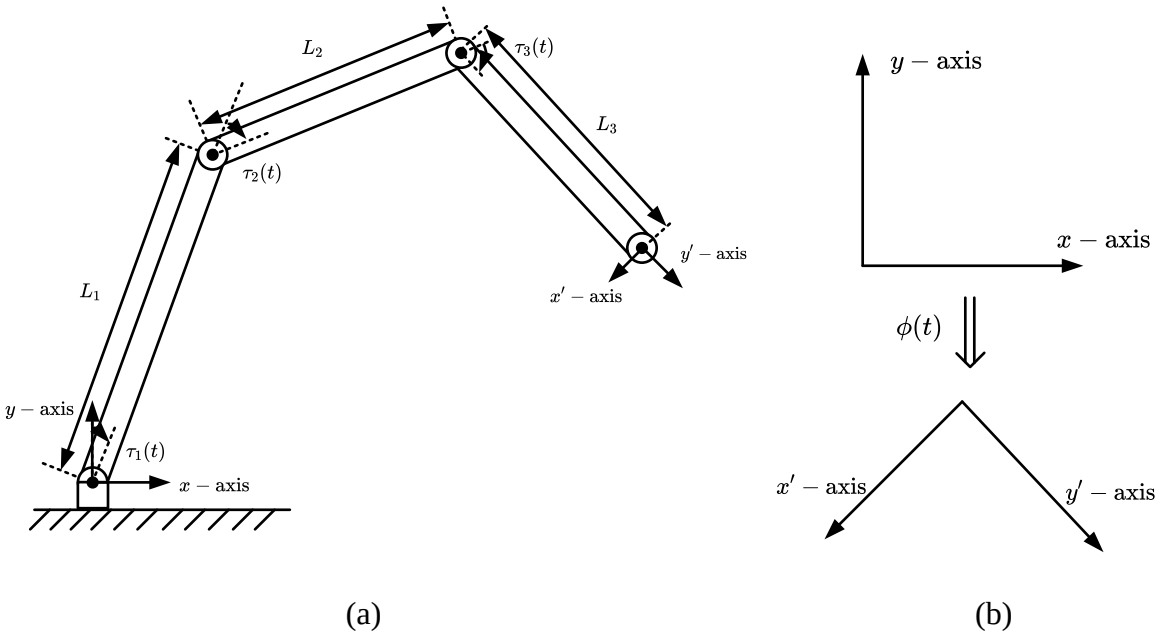


Figure 4.4: Three-link two-dimensional robotic arm model.

4.3.2 Knowledge-Assisted Problem Formulation

State

The state in the t-th time slot consists of two parts: the angles of the I joints and the Jacobian matrix of the real-world robotic arm, i.e., $\mathbf{s}_t = \{\mathscr{T}(t), \mathscr{J}(\mathscr{T}(t))\}$. In robotics, the Jacobian matrix is critical for analyzing and controlling the motion of robots. It characterizes the relationship between the velocity of the end effector and the velocities of all joints [155],

$$\frac{\Delta\mathscr{P}(t)}{\Delta t} = \mathscr{J}(\mathscr{T}(t))\frac{\Delta\mathscr{T}(t)}{\Delta t}, \tag{4.13}$$

where $\frac{\Delta\mathscr{P}(t)}{\Delta t}$ is the velocity of the end effector, and $\frac{\Delta\mathscr{T}(t)}{\Delta t}$ is the angular velocities of I joints. In the t-th time slot, the Jacobian matrix can be obtained from $\mathscr{T}(t)$ and the kinematic properties of the robotic arm, e.g., D-H parameters [155]. By multiplying Δt on both sides of (4.13), the relationship between $\Delta\mathscr{P}(t)$ and $\Delta\mathscr{T}(t)$ is expressed as follows,

$$\Delta\mathscr{P}(t) = \mathscr{J}(\mathscr{T}(t))\Delta\mathscr{T}(t), \tag{4.14}$$

where $\mathscr{J}(\mathscr{T}(t))$ shows how sensitive the modeling error of the end effector is to the errors of the I joints. Thus, we take the Jacobian matrix as one part of the state to improve the training efficiency of the DRL algorithm.

Let's take the three-link two-dimensional robotic arm as an example to show the Jacobian matrix [164]. With the example in Fig. 4.4(a), the forward kinematics in (4.8) can be expressed as follows,

$$\mathscr{P}(t) = \begin{bmatrix} l_{\mathrm{x,r}}(t) \\ l_{\mathrm{y,r}}(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} L_1 \cdot \cos\tau_1(t) + L_2 \cdot \cos\left(\tau_1(t)+\tau_2(t)\right) + L_3 \cdot \cos\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right) \\ L_1 \cdot \sin\tau_1(t) + L_2 \cdot \sin\left(\tau_1(t)+\tau_2(t)\right) + L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right) \\ \tau_1(t)+\tau_2(t)+\tau_3(t) \end{bmatrix}, \tag{4.15}$$

where L_1, L_2, and L_3 are the lengths of the three links, respectively. As shown in Fig. 4.4(b), $\phi(t)$ is the angle between x-axis and x'-axis in the clockwise direction. Then, the Jacobian matrix can be obtained by

$$\mathscr{J}\left(\mathscr{T}(t)\right) = \begin{bmatrix} \frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_1(t)} & \frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_2(t)} & \frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_3(t)} \\ \frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_1(t)} & \frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_2(t)} & \frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_3(t)} \\ \frac{\partial \phi}{\partial \tau_1(t)} & \frac{\partial \phi}{\partial \tau_2(t)} & \frac{\partial \phi}{\partial \tau_3(t)} \end{bmatrix} \tag{4.16}$$

where $\mathscr{J}\left(\mathscr{T}(t)\right)$ consists of all partial derivatives of $\mathscr{P}(t)$. Specifically, the first two rows of the matrix are related to the partial derivatives of the position coordinates, while the last row is related to the partial derivatives of the angle of the end effector which is shown in Fig. 4.4(b). Thus, each element is calculated by

$$\frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_1(t)} = -L_1 \cdot \sin\tau_1(t) - L_2 \cdot \sin\left(\tau_1(t)+\tau_2(t)\right) - L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.17}$$

$$\frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_2(t)} = -L_2 \cdot \sin\left(\tau_1(t)+\tau_2(t)\right) - L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.18}$$

$$\frac{\partial l_{\mathrm{x,r}}(t)}{\partial \tau_3(t)} = -L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.19}$$

$$\frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_1(t)} = -L_1 \cdot \sin\tau_1(t) - L_2 \cdot \sin\left(\tau_1(t)+\tau_2(t)\right) - L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.20}$$

$$\frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_2(t)} = -L_2 \cdot \sin\left(\tau_1(t)+\tau_2(t)\right) - L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.21}$$

$$\frac{\partial l_{\mathrm{y,r}}(t)}{\partial \tau_3(t)} = -L_3 \cdot \sin\left(\tau_1(t)+\tau_2(t)+\tau_3(t)\right), \tag{4.22}$$

$$\frac{\partial \phi}{\partial \tau_1(t)} = 1, \qquad \frac{\partial \phi}{\partial \tau_2(t)} = 1, \qquad \frac{\partial \phi}{\partial \tau_3(t)} = 1. \tag{4.23}$$

From the above description, we can see that the modeling error of the end effector is more sensitive to the error of the joint that is far away from the end effector and less sensitive to the error of the joint that is close to the end effector. The robotic arm in our prototype has more than

three joints and can move in a three-dimensional space. Thus, the Jacobian matrix could be more complicated than the two-dimensional robotic arm in Fig. 4.4.

The state, including joint angles and elements of the Jacobian matrix, needs to be normalized before feeding it into the neural network. We first find the maximum and minimum values of each joint angle and each element of the Jacobian matrix from the data set. Then, these values are employed to normalize the state within the range of $(0,1)$.

Action

The action to be taken in the t-th time slot includes the joints to be scheduled, $X(t)$, and the optimal prediction horizons $Z(t)$. Although the prediction horizon needs to be transmitted to the server, $Z(t)$ is an integer ranging from 1 to 500. Thus, the communication overhead for updating $Z(t)$ is negligible compared to the update of the joint angle with high precision. We denote the action by $\mathbf{a}_t = [\mathbf{a}_t^{[1]}, \mathbf{a}_t^{[2]}] = [X(t), Z(t)]$, where the action of the i-th joint is denoted by $\mathbf{a}_{t,i}^{[1]} = x_i(t)$ and $\mathbf{a}_{t,i}^{[2]} = z_i(t)$.

Instantaneous Reward and Cost

Given the state and the action taken in the t-th time slot, the instantaneous reward, denoted by $r(\mathbf{s}(t), \mathbf{a}(t))$, is the communication load reduction compared with a benchmark that all joints are scheduled in every time slot. The instantaneous cost $c(\mathbf{s}(t), \mathbf{a}(t))$ is set to $\mathbf{e}(t)$ in (4.9).

Table 4.1: Hyperparameters Settings

Network Name	Function	Input & Output	Layer Name	Units	Activation
Policy Network	Input 1	7 - 32	Linear	32	ReLU
	Input 2	42 - 32	Linear	32	ReLU
	Integration	32 - 64	Linear	64	ReLU
	Output 1	64 - 32	Linear	32	ReLU
		32 - 7	Linear	32	/
	Output 2	64 - 32	Linear	32	ReLU
		32 - 7	Linear	32	/
Critic Network	Input 1	7 - 32	Linear	32	ReLU
	Input 2	7 - 32	Linear	32	ReLU
	Output	32 - 64	Linear	64	ReLU
		64 - 32	Linear	32	ReLU
		32 - 1	Linear	32	/
Cost Network	Input 1	7 - 32	Linear	32	ReLU
	Input 2	7 - 32	Linear	32	ReLU
	Output	32 - 64	Linear	64	ReLU
		64 - 32	Linear	32	ReLU
		32 - 1	Linear	32	/

Policy

The policy is represented by a neural network, $\pi_\theta(\mathbf{s}_t)$, where θ represents the training parameters. The network consists of multiple fully connected layers as shown in Fig. 4.5 and the settings of DRL networks are shown in Table 4.1. In our study, the inputs to the policy networks include two different states: the angles of joints and the Jacobian matrix of the real-world robotic arm. The raw data of a joint angle has complete information and requires a complex neural network for feature extraction. The Jacobi matrix provides information that has been processed based on domain knowledge, and we can use a simple neural network for feature extraction. To handle different types of input information, we designed the two-branch neural network. Meanwhile, the two branches are designed to optimize two types of actions separately, i.e., the scheduling of a joint and the prediction horizon.

Specifically, the first two layers are designed for feature extraction, where the input denoted by $\{\mathscr{T}(t), \mathscr{J}(\mathscr{T}(t))\}$ is transformed into a more compact and informative representation that captures the underlying patterns. Then, we decouple the neural network into two parallel branches. The first branch is for the scheduling policy, $\pi_\theta^{[1]}$, and the second branch is for the policy of optimizing prediction horizons, $\pi_\theta^{[2]}$. After that, two branches are concatenated in the final linear layer. Followed by the Softmax activation function, the distribution of two actions, i.e., $\rho_t^{[1]}$ and $\rho_t^{[2]}$ are generated. Finally, two actions, i.e., $\mathbf{a}_t^{[1]}$ and $\mathbf{a}_t^{[2]}$ are sampled from $\rho_t^{[1]}$ and $\rho_t^{[2]}$, respectively.

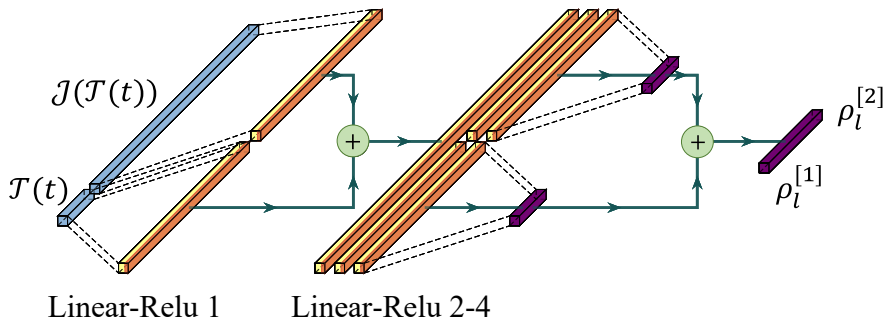
Specifically, $\pi_\theta^{[1]}$ maps the state $\mathbf{s}_t$ to the distribution of $\mathbf{a}_{t,i}^{[1]}$, denoted by $\rho_t^{[1]} \in \mathbb{R}^{2\times I}$. The i-th column of $\rho_t^{[1]}$ is defined as follows,

$$\rho_{t,i}^{[1]} \triangleq \begin{pmatrix} \Pr\{a_{t,i}^{[1]} = 1\} \\ \Pr\{a_{t,i}^{[1]} = 0\} \end{pmatrix}. \tag{4.24}$$

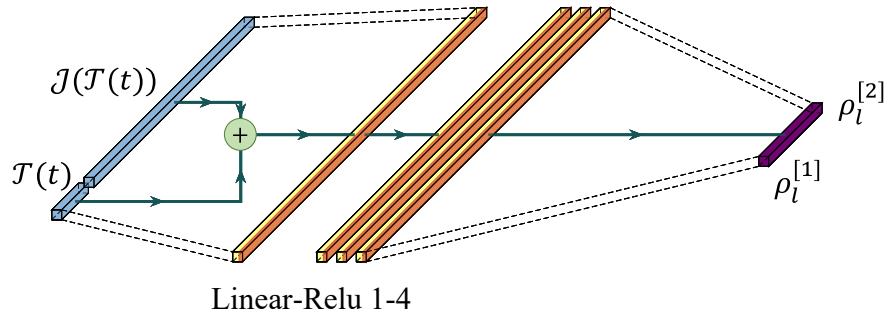
Similarly, $\pi_\theta^{[2]}$ maps the state $\mathbf{s}_t$ to the distribution of $\mathbf{a}_{t,i}^{[2]}$, denoted by $\rho_t^{[2]} \in \mathbb{R}^{H\times I}$. The i-th column of $\rho_t^{[2]}$ is defined as follows,

$$\rho_{t,i}^{[2]} \triangleq \left[\Pr\{a_{t,i}^{[2]} = 1\}, \Pr\{a_{t,i}^{[2]} = 2\}, ..., \Pr\{a_{t,i}^{[2]} = H\}\right]^{\mathrm{T}}. \tag{4.25}$$

Once the distributions are obtained, $\mathbf{a}_{t,i}^{[1]}$ and $\mathbf{a}_{t,i}^{[2]}$ can be sampled from (4.24) and (4.25), respectively. Here, the probability of each action being sampled is based on the weight located in the corresponding elements [172]. The policies of different joints are represented by DNNs with the same structure. If there are more joints and sensors, they can reuse the DNN and fine-tune the parameters with few-shot learning. In this way, we can address the scalability issue.



(a) Two-branch Neural Network.



(b) Fully-Connected Neural Network.

Figure 4.5: Structures of neural networks: (a) Two-branch neural network, (b) fully-connected neural network.

Long-Term Reward and Cost

Following the policy π_θ, the long-term reward and long-term cost are defined as

$$R^{\pi_\theta} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}(t), \mathbf{a}(t)], \tag{4.26}$$

$$C^{\pi_\theta} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(\mathbf{s}(t), \mathbf{a}(t)], \tag{4.27}$$

where γ is the discount factor [160]. To estimate the long-term reward and long-term cost, we can use the state-value function or the state-action value function. The state-value function and the state-action-value function of the long-term reward are defined as

$$V_r^{\pi_\theta}(\mathbf{s}) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}(t), \mathbf{a}(t)) \mid \mathbf{s}_0 = \mathbf{s},\ \pi_\theta], \tag{4.28}$$

$$Q_r^{\pi_\theta}(\mathbf{s}, \mathbf{a}) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}(t), \mathbf{a}(t)) \mid \mathbf{s}_0 = \mathbf{s},\ \mathbf{a}_0 = \mathbf{a},\ \pi_\theta], \tag{4.29}$$

respectively. The advantage function is given by $A_r^{\pi_\theta}(\mathbf{s}, \mathbf{a}) = Q_r^{\pi_\theta}(\mathbf{s}, \mathbf{a}) - V_r^{\pi_\theta}(\mathbf{s})$. Like the long-term reward, the state-value function and the state-action-value function of the long-term cost are

defined as

$$V_c^{\pi_\theta}(\mathbf{s}) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(\mathbf{s}(t), \mathbf{a}(t)) \mid \mathbf{s}_0 = \mathbf{s},\ \pi_\theta], \tag{4.30}$$

$$Q_c^{\pi_\theta}(\mathbf{s}, \mathbf{a}) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(\mathbf{s}(t), \mathbf{a}(t)) \mid \mathbf{s}_0 = \mathbf{s},\ \mathbf{a}_0 = \mathbf{a},\ \pi_\theta], \tag{4.31}$$

respectively. The advantage function is given by $A_c^{\pi_\theta}(\mathbf{s}, \mathbf{a}) = Q_c^{\pi_\theta}(\mathbf{s}, \mathbf{a}) - V_c^{\pi_\theta}(\mathbf{s})$.

Modeling Accuracy Constraint

To guarantee the long-term modeling accuracy constraint, a straightforward approach is to evaluate C^{π_θ} by using $V_c^{\pi_\theta}(\mathbf{s})$ or $Q_c^{\pi_\theta}(\mathbf{s}, \mathbf{a})$. Note that the average long-term cost may not be applicable for mission-critical tasks in the Metaverse. For example, in haptic communications, users cannot recognize errors below a certain threshold, known as JND [173]. For mission-critical tasks, we propose to use CVaR of $Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ as the KPI. CVaR is a well-known risk measure used in financial portfolio analysis that depicts the cost in the tail of the risk distribution [174]. The expression of CVaR of $Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ is given by

$$\text{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)] = \min_{v \in \mathscr{R}} \left(v + \frac{1}{1-\alpha} \mathop{\mathbb{E}}_{\mathbf{s}_t, \mathbf{a}_t \sim \pi_\theta} \left\{ [Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t) - v]^+ \right\} \right), \tag{4.32}$$

where $(x)^+ = \max(x, 0)$. $\alpha \in (0, 1)$ is the confidence level, and $Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ is equal to the average of the worst-case α-fraction of losses under optimal conditions [175].

Problem Formulation

The goal is to find the optimal policy π_θ^* that maximizes the long-term reward R^{π_θ} subject to the constraint on CVaR of the long-term cost C^{π_θ}. Thus, the problem can be formulated as follows:

$$\pi_\theta^* = \arg\max_\theta Q_r^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t) \tag{4.33}$$

$$\text{s.t.} \quad \text{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)] \leq \frac{\Gamma_c}{1-\gamma}, \tag{4.33a}$$

where Γ_c is the modeling error depending on the requirements of specific tasks in the Metaverse.

4.4 C-PPO Algorithm

To guarantee the modeling accuracy constraint, we develop a C-PPO algorithm by integrating PPO and CVaR into the CRPO algorithm, which is a safe reinforcement learning algorithm with convergence guarantee [160]. The basic idea of the CRPO algorithm is to maximize the long-term reward when the constraint is satisfied and minimize the long-term cost when the constraint is violated.

The details of the C-PPO algorithm can be found in Algorithm 1. In the t-th step, we first update the threshold of CVaR according to the current policy by the gradient descent,

$$v^{(k+1)} = v^{(k)} - \beta \frac{\Delta \text{CVaR}_\alpha(v^{(k)})}{\Delta v^{(k)}}, \tag{4.34}$$

where β is the learning rate and it takes K steps of gradient descent to update the threshold, $k = 1, ..., K$. Then, we validate whether the constraint can be satisfied or not. If the constraint in (4.33a) is satisfied, we maximize $\mathscr{L}_r(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta)$ which is obtained by substituting $A_r^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ into (4.11). Otherwise, we minimize $\mathscr{L}_c(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta)$ defined as follows,

$$\begin{aligned} \mathscr{L}_c(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta) = \min \Bigg(& \frac{\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\theta_t}(\mathbf{a}_t \mid \mathbf{s}_t)} A_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t), \\ & \text{clip}\left(\frac{\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\theta_t}(\mathbf{a}_t \mid \mathbf{s}_t)}, 1-\varepsilon, 1+\varepsilon, \right) A_c^{\pi_{\theta_t}}(\mathbf{s}_t, \mathbf{a}_t) \Bigg), \end{aligned} \tag{4.35}$$

where $A_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ is obtained by Generalized Advantage Estimate (GAE) [171], [176]. With C-PPO, the parameters are updated according to the following expression,

$$\theta_{t+1} = \begin{cases} \theta_t + \alpha \nabla_\theta \mathscr{L}_r(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta), & \text{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)] \leq \frac{\Gamma_c}{1-\gamma}, \\ \theta_t - \alpha \nabla_\theta \mathscr{L}_c(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta), & \text{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)] > \frac{\Gamma_c}{1-\gamma}. \end{cases} \tag{4.36}$$

It is worth noting that the policy gradient and the CVaR gradient can be updated with different learning rates. To guarantee a stable CVaR constraint when performing the policy gradient, we update the threshold of CVaR with a higher frequency.

4.5 Prototype Design and Performance Evaluation

In this section, we demonstrate our cross-system prototype design as shown in Fig. 4.6. Based on the prototype, we first evaluate the effectiveness of the proposed cross-system design framework and then compare the performance with different benchmarks.

4.5.1 Prototype Design

Real-World Robotic Arm

We adopt an industrial-grade robotic arm system, Franka Emika Panda [153], in our prototype. The robotic arm has seven DoF and can achieve up to 2 m/s end effector speed and ± 0.1 mm repeatability. In our prototype design, we use five DoF of the real-world robotic arm. The trainer wearing optical markers controls the robotic arm via the state-of-the-art motion capture system with six motion cameras, OptiTrack Prime-13 [177]. The reason why we use OptiTrack is because

Algorithm 2 C-PPO

Input: initial the parameters of neural network including policy network θ_0, initial state $\mathbf{s}_0$, step length β

for $t = 0, 1, 2, \ldots, T-1$ **do**
  Observe $\mathbf{s}_t$ and generate action from current policy $\pi_{\theta_t}([\mathbf{a}_t^{[1]}, \mathbf{a}_t^{[2]}] \mid \mathbf{s}_t)$
  Transmit the packets based on action $\mathbf{a}_t^{[1]}$
  Reconstruct the trajectory based on received packets by (4.1)
  Predict the trajectory based on action $\mathbf{a}_t^{[2]}$ by (4.2)
  Store state $\mathbf{s}_t$, action $\mathbf{a}_t$, reward r_t, cost c_t, and next state $\mathbf{s}_{t+1}$
  for $k = 1, 2, \ldots, K$ **do**
    Update $\mathrm{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)]$ by (4.34)
  end for
  Compute the advantage function $A_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ and $A_r^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)$ based on (4.11), (4.35)
  if $\mathrm{CVaR}_\alpha[Q_c^{\pi_\theta}(\mathbf{s}_t, \mathbf{a}_t)] \leq \frac{\Gamma_c}{1-\gamma}$ **then**
    Take one-step policy update towards maximizing $\mathscr{L}_r(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta) : \theta_t \to \theta_{t+1}$
  else
    Take one-step policy update towards maximizing $\mathscr{L}_c(\mathbf{s}_t, \mathbf{a}_t, \theta_t, \theta)$: $\theta_t \to \theta_{t+1}$
  end if
end for

Output: Optimal policy π_θ^*

1) It is a real-time motion capture system with high accuracy and low latency. This real-time high-fidelity tracking is essential for maintaining accurate digital twins in the Metaverse and for evaluating the performance of our task-oriented, cross-system design framework. The use of OptiTrack also allows us to quantify the task-oriented KPI, i.e., average tracking error between the real-world robotic arm and its digital model in the Metaverse.

2) The OptiTrack motion tracking system can be scaled to accommodate various tracking volumes, from small studio setups to large outdoor environments. This scalability makes it versatile and adaptable for different types of motion capture objects in the Metaverse.

3) Our proposed framework is not limited to the use of Optitrack as a motion capture device. Flexible positioning and motion tracking systems can be used based on the practical accuracy demands of Metaverse applications.

The robotic arm receives the target pose from the motion capture system and then maps the pose to joint angles. After that, the robotic arm applies a proportional-integral-derivative method [154] for control, which converts the joint angles to a series of commands. Meanwhile, built-in sensors in the robotic arm measure joint angles, angular velocities, and inertial torque of each joint [153].

Virtual Robotic Arm in the Metaverse

We establish the Metaverse in the Nvidia Isaac Gym [178], a cutting-edge robotics simulation engine that uses state-of-the-art algorithms and physics engines to simulate the movement and

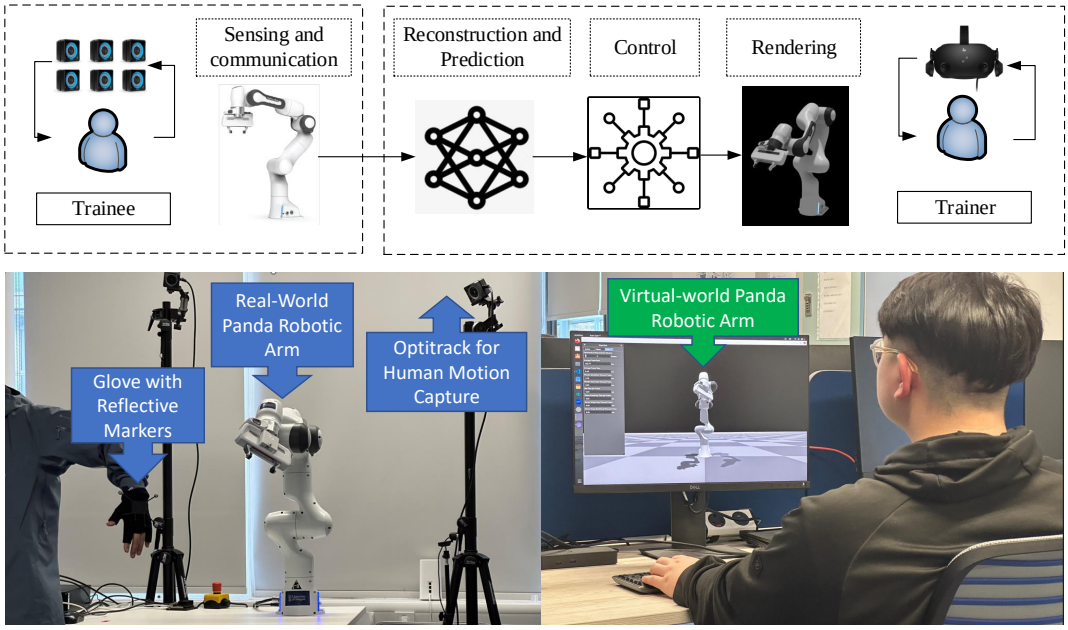


Figure 4.6: Illustration of our prototype system. 1) Bottom left photo: A real-world robotic arm is controlled by a human trainee, 2) Bottom right photo: A digital model of the robotic arm in the Metaverse is rendered and presented to a trainer, 3) The diagram on the top of the two photos shows the system functions implemented at the two sides.

behavior of robots in various environments. Meanwhile, we simulate the communication system between the real-world robotic arm and the Nvidia Isaac Gym by introducing a Gaussian-distributed communication latency. Its mean and standard deviation are set to 10 ms and 1 ms, respectively.

4.5.2 System Setup

Parameters of the Prototype

For the prototype design, five joint angles of the real-world robotic arm are controlled by the trainee and measured by built-in sensors in each time slot. The measured data of the real-world and virtual-world robotic arms are recorded in the CSV format file. For the training process of the predictor, *informer*, we set the prediction input length W_p to 2000 ms and output length H to 500 ms. In Nvidia Issac Gym, we set the control interval N_c to 2 ms. The frequency at which the robotic arm interacts with the virtual environment is 500 Hz. The method for calculating the Jacobian Matrix can be found in Appendix B. The default parameters of the prototype and learning algorithm are listed in Table 4.5.1, unless otherwise specified.

Prototype Setup	
Parameters	**Values**
Duration of time slot	1 ms
Number of joints I	5
Input length of reconstruction function W_l	2000 ms
Input length of prediction function W_p	2000 ms
Prediction horizon of prediction function H	500 ms
Control interval N_c	2 ms
Refresh rate of image $1/N_r t_s$	60 Hz
Experimental time	5×10^4 ms
Weighting coefficient of position ω_1	0.5
Weighting coefficient of orientation ω_2	0.5
Learning Setup	
Parameters	**Values**
Learning rate of actor-network	3×10^{-4}
Learning rate of critic network	3×10^{-4}
Learning rate of cost network	3×10^{-4}
Batch size	256
Discount factor γ	0.99
Clip ratio in the loss functions of C-PPO ε	0.2
Total steps for updating CVaR K	500
Learning rate of constraint β	2×10^{-3}
Confidence level of CVaR $1 - \alpha_c$	0.95

Table 4.2: System Parameters for Performance Evaluation

Benchmarks

We build our C-PPO algorithm and four benchmarks in the well-known DRL library *Stable-Baselines3* [176]. The legends of the benchmarks are "W2B", "WJM", "WCVaR", and "WDK", respectively. (a) In W2B, the two-branch neural network is replaced with a fully-connected neural network; (b) In WJM, the Jacobian matrix is not included in the state; (c) In WCVaR, the CVaR of the modeling error is replaced with the average modeling error in the constraint; (d) WDK is a simplified C-PPO without using any domain knowledge (i.e., the two-branch neural network, Jacobian matrix, and CVaR of the modeling error). With the above benchmarks, we will illustrate the impact of different types of domain knowledge on the performance of our C-PPO algorithm.

4.5.3 Evaluation of C-PPO Algorithm

As shown in Figure 4.6, the performance of C-PPO is evaluated by average packet rate, average modeling error, and average CVaR. We train C-PPO with 800 episodes to show the trends in performance changes. The average packet rate at the start point is defined as a baseline. The average packet rate reaches 8 packets/second, which saves 83.3% communication load than

the baseline. In addition, optimizing under the restriction of the constraint CVaR, the average tracking error of the digital model achieves 0.0281, which is lower than the tracking error of the baseline (0.0347), as shown in Figure 4.7(b). Meanwhile, the CVaR fluctuates around the constraint bound. The results show that C-PPO converges after 250 training episodes. Meanwhile, constraints and the average tracking error fluctuate slightly throughout training. In particular, C-PPO is stable and effective, since it performs consistently better in all ten training repetitions. It is also worth noting that although the prediction horizon needs to be transmitted to the server, $Z(t)$ is an integer ranging from 1 to 500. Thus, the communication overhead for updating $Z(t)$ is negligible compared to the update of the joint angle with high precision.

Ablation Study of C-PPO Algorithm

The performance of the C-PPO and the four benchmarks are illustrated in Fig. 4.6. In general, our C-PPO achieves the best performance in terms of convergence time, average packet rate, and average modeling error. The results in Fig. 4.7(a) show that the C-PPO can reduce the required average packet rate by around 50% compared to the benchmark without domain knowledge, WDK (from 17 packets/second to 8 packets/second). By comparing C-PPO with WJM, we can see that the Jacobian matrix can reduce the convergence time by 50% (from 800 episodes to 400 episodes). In Fig. 4.7(b), we evaluate the average modeling errors achieved by different algorithms. From C-PPO and WCVaR, we can observe that by using CVaR as the metric of the modeling error, the average modeling error is reduced from 0.041 to 0.032 (around 20% reduction), where the modeling error is defined in (4.9). The results 4.6(c) show that C-PPO, W2B, and WJM can guarantee the constraint on CVaR of the modeling error, that is, the dashed horizontal line. The other two benchmarks do not consider CVaR, and hence are not shown in this figure. From the performance of W2B in all these figures, we can see that the two-branch neural network converges faster than the fully-connected neural network and achieves better performance in terms of the average packet rate and average modeling error.

4.5.4 Validation of Cross-System Design Framework

Dynamic scheduling in C-PPO

In Fig. 4.7(a), we provide an example to show how the proposed C-PPO algorithm changes the packet rates according to joint angles. The packet rates are represented by the grayscale intensity. As the grayscale intensity increases from white to black, the packet rates increase from 0 to 23 packets/second. The results imply that packet rates are correlated with fluctuations in joint angles. Besides, the joint that is far away from the end effector has higher average packet rates than the joint that is close to the end effector. This is because the modeling error of the end effector is less sensitive to modeling errors of the joints that are closer to it.

Dynamic prediction horizon in C-PPO

The effect of $Z(t)$ has already been demonstrated in the existing literature [179], [180]. The latency in the communication systems is stochastic, so we need to adjust the prediction horizon to compensate for the communication latency. In this way, we can reduce the modeling error in the Metaverse. In addition, we also provide the diagram of $Z(t)$ along with the trajectories of the real-world robotic arm. As shown in Fig. 4.7(b) in the response letter, the alteration of the prediction horizon is depicted through variations in grayscale intensity, where darker intensities correspond to higher packet rates across the prediction horizon that spans from the $1st$ to the $500th$ packet. The results of the study indicate a notable correlation between the prediction horizon and the fluctuations observed in the joint angles. Moreover, it is observed that joints situated farther from the end effector exhibit longer predictive spans compared to those located in closer proximity to the end effector. This discrepancy is attributed to the reduced sensitivity of the end effector's modeling errors to perturbations in the joints near it.

CVaR

The CCDF of the long-term cost is presented in Fig. 4.8. The results show that with the probability of 98.5%, the long-term cost is below the required threshold, which is set to 25 in the experiment. In addition, the probability (98.5%) is higher than the confidence level (95%). The result also indicates that the proposed C-PPO can significantly reduce the tail probability (i.e., the probability that the long-term cost is higher than the required threshold) of the long-term cost.

Performance Comparison

We compare our proposed cross-system design framework with a baseline framework: there is no prediction, and all joints transmit data packets in every time slot. In Fig. 4.9, we show the trajectories of the real-world robotic and two digital models obtained from our cross-system design framework and the baseline framework. The results show that with prediction, the cross-system design framework can model the virtual-world robotic arm in a timely manner. Without prediction, the user can recognize the modeling error caused by communication latency. In Fig. 4.10, we further illustrate the CCDF of modeling errors in the two frameworks. The results demonstrate that the cross-system design framework outperforms the baseline framework in terms of the tail distribution of the modeling error. Besides, with the baseline framework, all the joints transmit packets in every time slot. The packet rate is 5000 packets/s, which is much higher than the cross-system design framework.

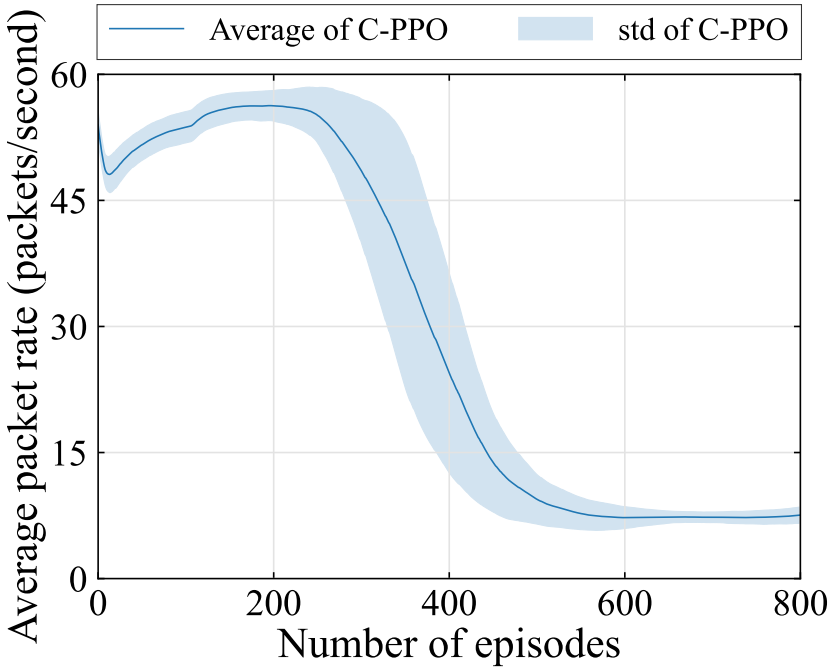
4.5.5 Discussion of Error Impact on Different Scenarios

The error of a robotic arm operation can significantly impact various application scenarios, including welding and other industries. For example, in welding applications, accuracy is crucial

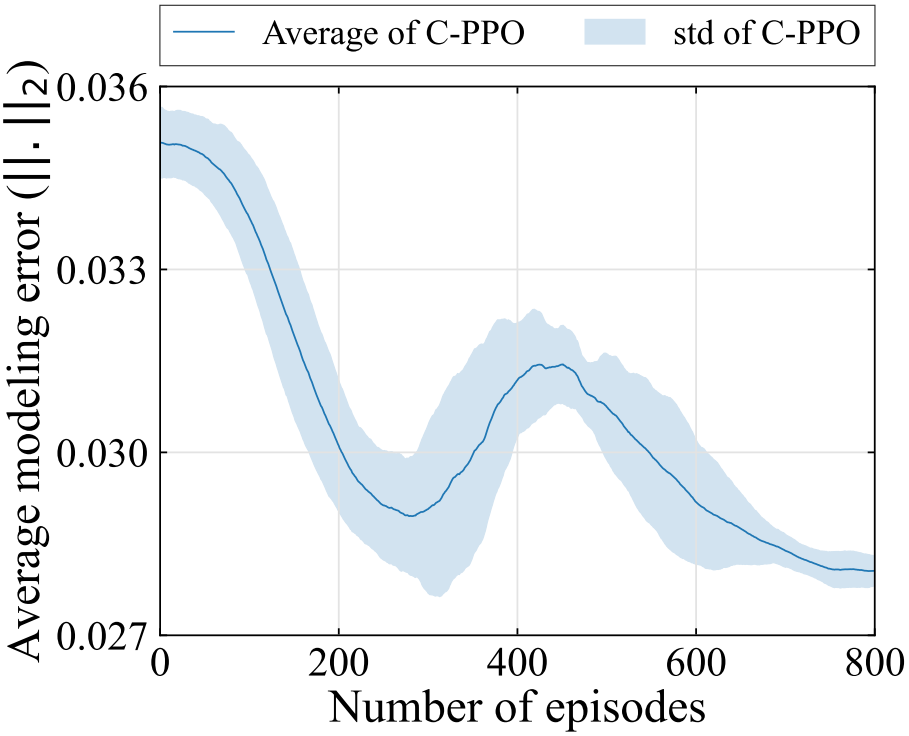
to ensure precise and consistent welds. A highly accurate robotic arm can follow predefined paths with minimal deviation, resulting in higher-quality welds and better structural integrity. The error tolerance may be very low, often in the range of ±0.1 mm to ±0.5 mm [181]. In Medical applications such as robotic-assisted surgery, the accuracy of operation is also critical, where accurate operations of robotic arms can assist surgeons in performing intricate tasks with minimal invasiveness, reducing the risk to patients. The error tolerance is extremely low, typically within the range of sub-millimeters, such as ±0.1 mm [182]. In warehouse automation for picking and placing items, the acceptable error tolerance may vary based on the size and nature of the products, but it could be in the range of ±1 cm to ±5 cm [183]. In our experiment setup.In our experiments, our mean square error is set to the range of automatic sorting and packing in an automated warehouse storage scenario, which illustrates the effectiveness of our algorithm. In addition, our designed framework has strong generalizability and can be used to set different accuracy constraints according to the different application scenarios described above.

4.6 Conclusions

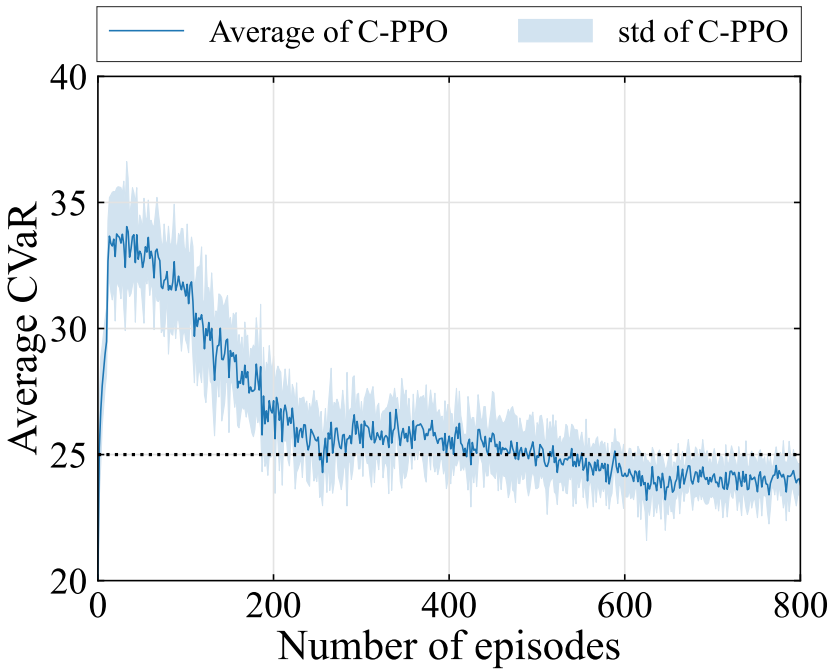
In this work, we established a task-oriented cross-system design framework to minimize the required packet rate to meet a constraint on the modeling error of a robotic arm in the Metaverse. To optimize the scheduling policy and the prediction horizons, we developed a C-PPO algorithm by integrating domain knowledge into the PPO. A prototype was built to evaluate the performance of the C-PPO and the cross-system design framework. Experimental results showed that the domain knowledge helps reduce the required packet rate and the convergence time by up to 50%, and the cross-design framework outperforms a baseline framework in terms of the required packet rate and the tail distribution of the modeling error.



(a) Average packet rate in each episode.

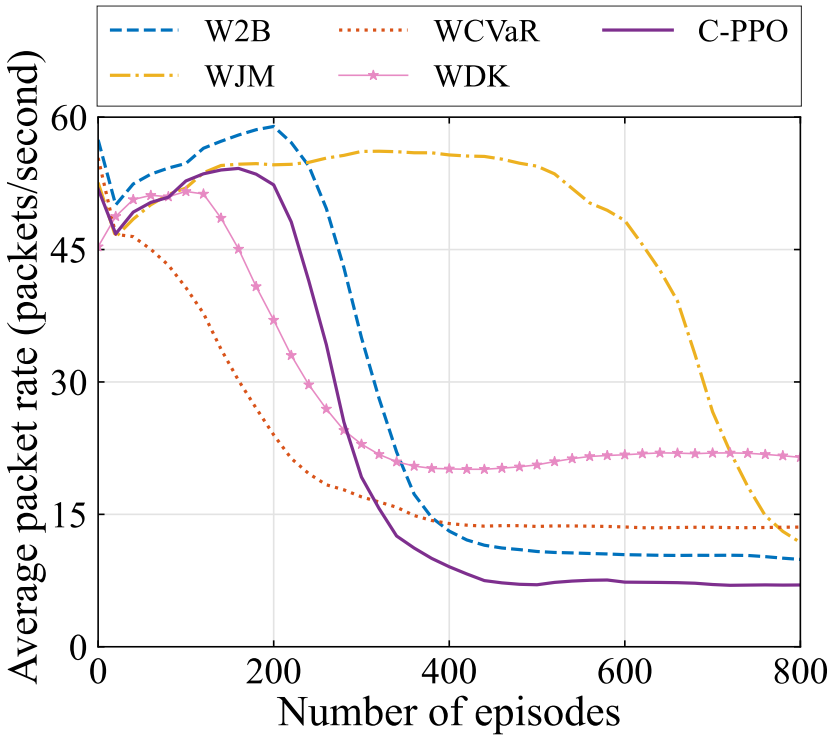


(b) Average modeling error in each episode.

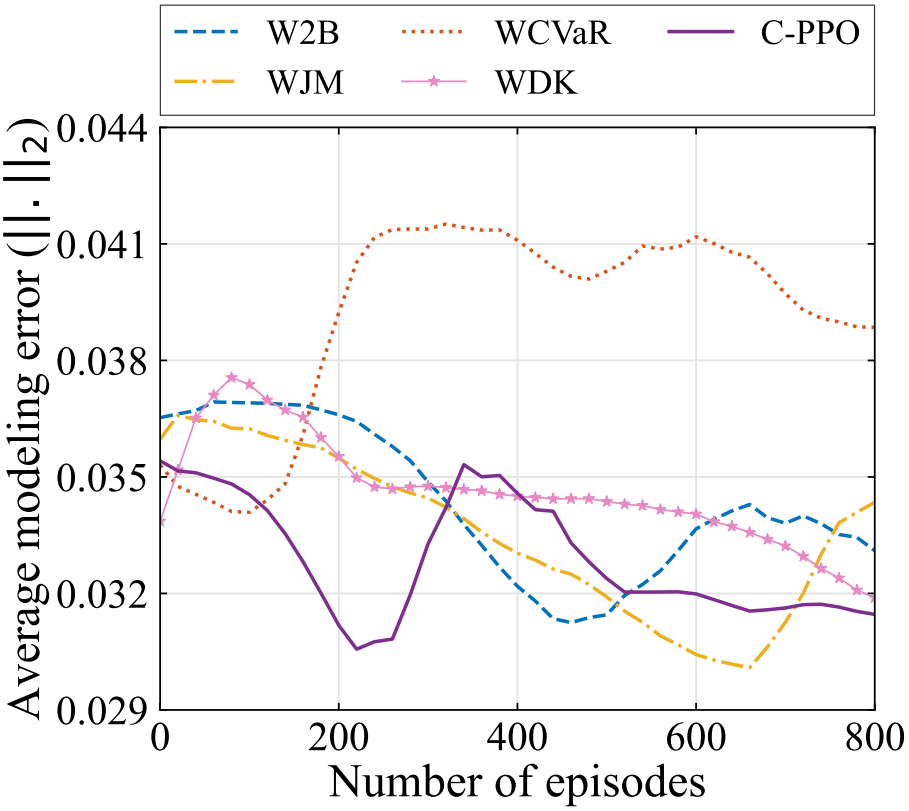


(c) Average value of CVaR in each episode.

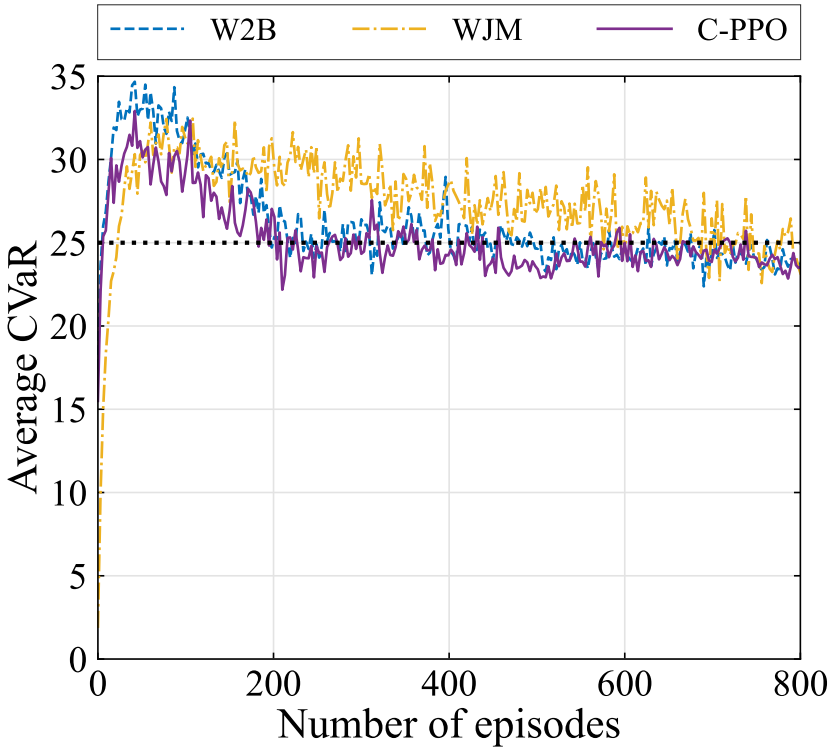
Figure 4.6: Performance Evaluation.



(a) Average packet rate in each episode.

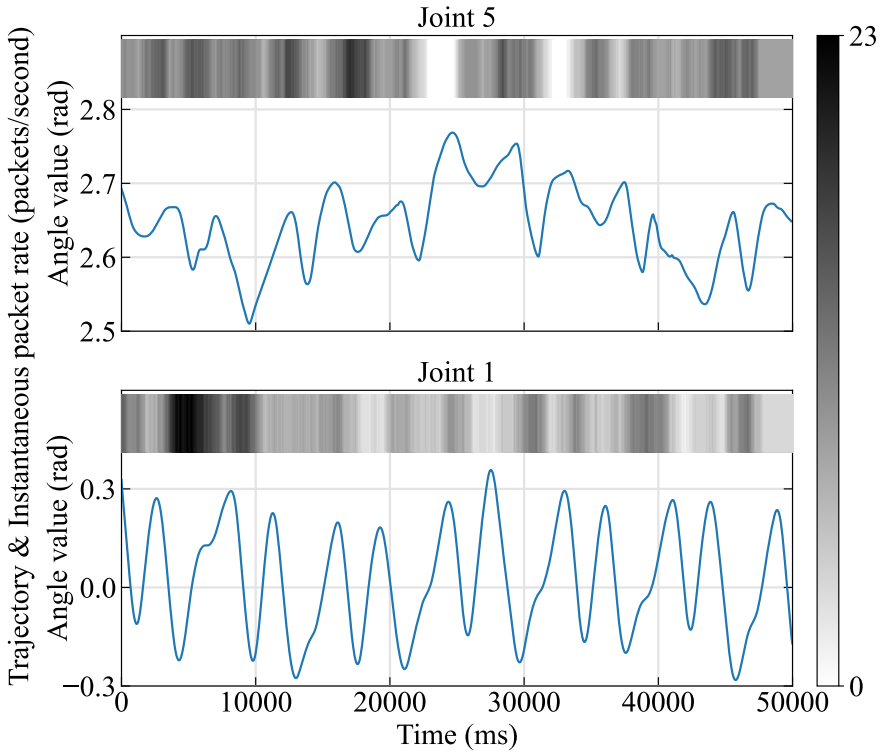


(b) Average modeling error in each episode.

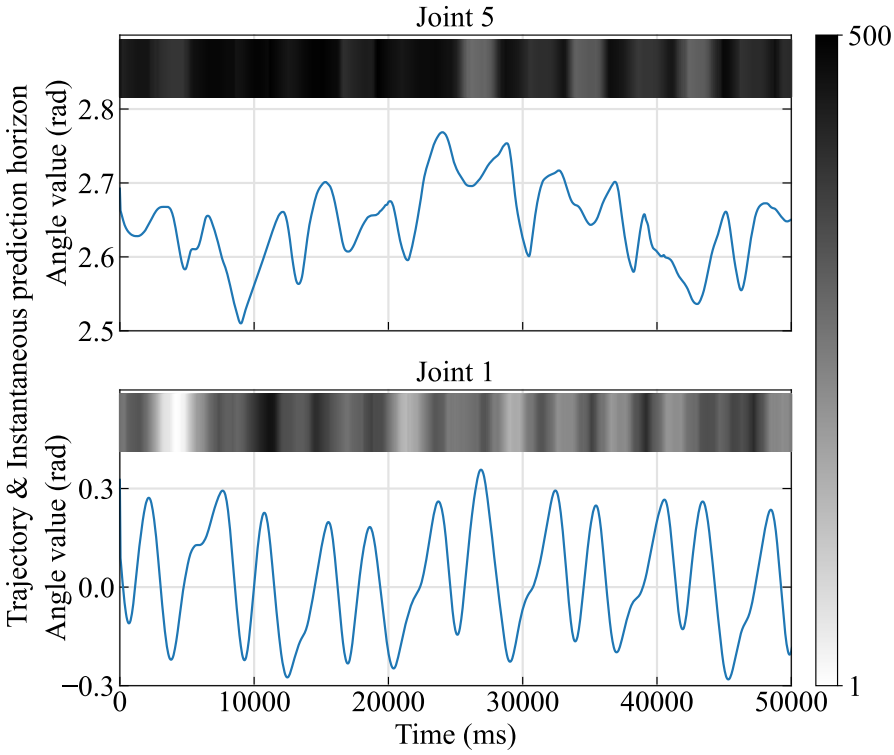


(c) Average value of CVaR in each episode.

Figure 4.6: Ablation Study.



(a) Trajectories and instantaneous packet rates



(b) Trajectories and instantaneous prediction horizon

Figure 4.7: Trajectories, instantaneous packet rates and instantaneous prediction horizon of two joints, where joint 5 is close to the end effector, and joint 1 is close to the base of the robotic arm.

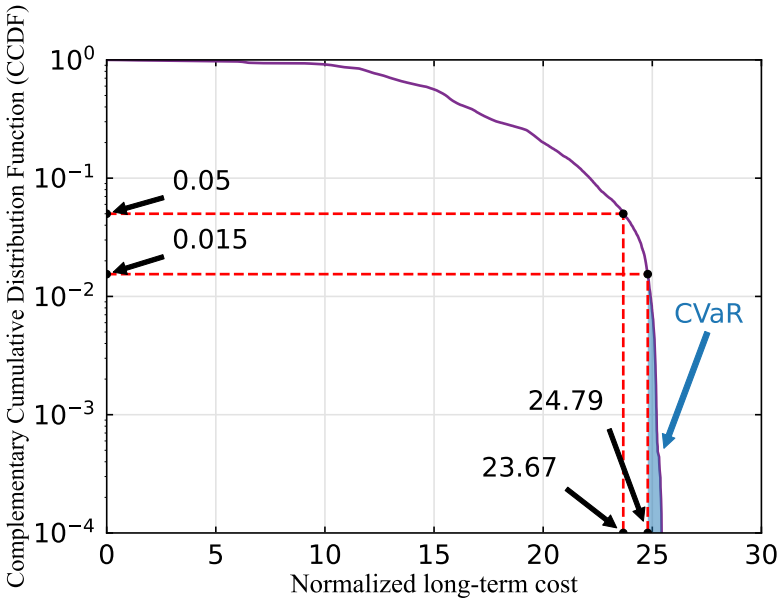


Figure 4.8: CCDF of the long-term cost.

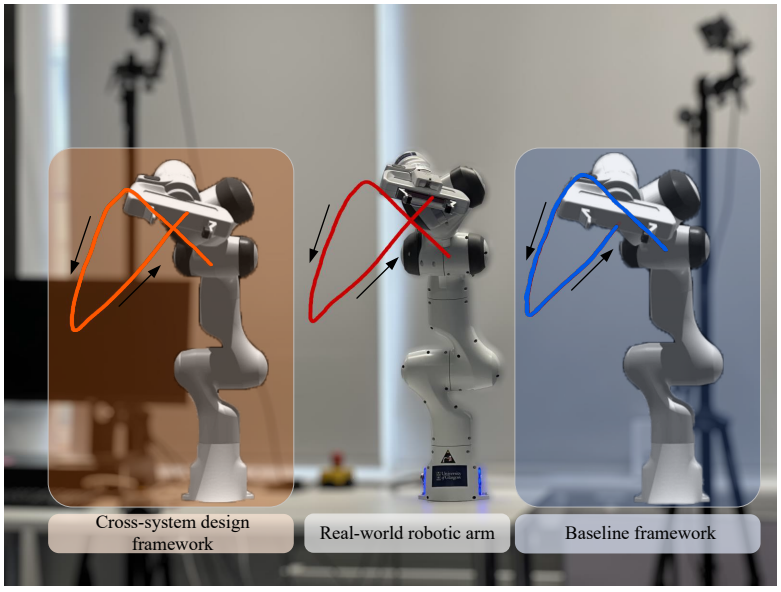


Figure 4.9: Comparison among the trajectories of the real-world robotic arm (at the centre) and its digital models (on two sides, the left one is designed by the proposed framework while the right one is implemented by the baseline framework without cross-system design).

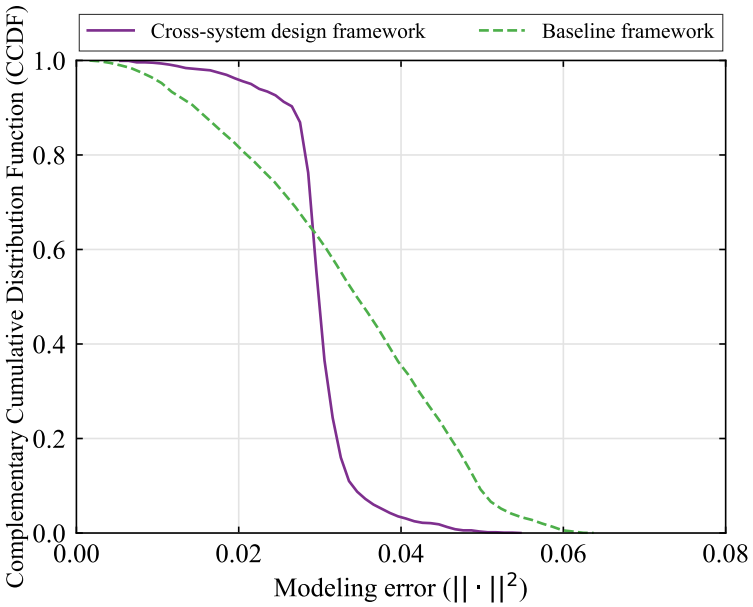


Figure 4.10: Modeling errors of the proposed cross-system design framework and the baseline framework.

Chapter 5

Conclusions and Open Issues

5.1 Conclusions

In this thesis, a transformative potential to reshape social interactions in the post-pandemic era of the Metaverse was comprehensively depicted. Despite its ambitious objectives, the realization of the Metaverse's comprehensive vision remains intricate due to the existing uncertainties surrounding methodologies and technologies required for its full implementation. Based on this, this thesis contributed to the advancement of the Metaverse concept by establishing a comprehensive framework for its development. The groundwork was laid through the identification of three foundational infrastructure pillars: Human-Computer Interaction (HCI), sensing and communication systems, and network architectures. These pillars form the bedrock upon which the progression toward the Metaverse is built.

The research journey progressed further by delineating a roadmap with distinct stages, each featuring applications that reflect the Metaverse's transformative potential. Crucially, the study explored the pivotal role of Computer Vision in realizing the Metaverse, accentuating its current state-of-the-art and outlining future directions for exploration. To address the challenge of constructing a functional Metaverse, a novel design methodology called task-oriented cross-system design was introduced. The methodology was demonstrated through concrete examples, where optimization efforts were directed towards achieving synchronization between real-world devices and digital models in the Metaverse. In these instances, the integration of domain knowledge proved instrumental, enabling the development of deep reinforcement learning algorithms that effectively balanced tracking accuracy with communication load.

Furthermore, the thesis extended its focus to encompass a broader scope involving multiple modules, including sensing, communications, prediction, control, and rendering. By adopting an advanced reinforcement learning algorithm, PPO, and incorporating domain knowledge, the task-oriented cross-system design was elevated through the creation of a C-PPO algorithm. The experimental validation of this framework demonstrated its prowess in reducing the convergence

time and required packet rates, confirming its superiority over baseline frameworks in terms of modeling accuracy and efficiency.

In essence, this research underscored the complex yet promising landscape of the Metaverse. By offering a structured approach to tackle its multifaceted challenges, from foundational infrastructure to application-specific optimization, the study contributes significantly to the collective understanding and eventual realization of the Metaverse's revolutionary potential. However, as a nascent concept, the journey towards the Metaverse's full realization is ongoing, inviting further research and collaboration to address the remaining uncertainties and unlock the transformative benefits it promises to bring to society.

5.2 Open Issues

5.2.1 Multi-tier Architecture for the Metaverse

The 6G network should be capable of achieving large-scale multi-tier diverse KPIs in the Metaverse by leveraging distributed computing, storage, and communication resources in local devices, edge servers, and central servers [8]. A hierarchical communication network architecture is required for applications such as mining automation, monitoring of forest fires, and remote control, where real-time feedback is essential. This architecture involves the synchronization of edge servers to infer the specific feedback, which allows users to interact with the edge with a satisfied experience [184]. However, how to quickly and effectively schedule different resources for different tasks in the Metaverse needs further investigation.

5.2.2 Integrated Sensing and Communications for the Metaverse

The task-oriented cross-system design between sensing and communications - ISAC - is one of the potential enablers in the future [11]. By utilizing the same radio frequency band and hardware resource, ISAC has the potential to access wider frequency bands and reduce the system's cost. This makes ISAC suitable for Metaverse applications. For example, ISAC can be used to track the movements of users in real-time and translate their actions and gestures to the corresponding actions in the virtual environment [185]. This provides a more natural and immersive user experience compared to traditional input devices such as keyboards or joysticks. In addition, combined with state-of-art technologies, such as massive MIMO and Intelligent Reflecting Surface (IRS) [186], ISAC can provide ultra-high bandwidth for AR/VR users. However, ISAC still faces many challenges for Metaverse applications, such as energy efficiency, spectrum management, and interference management, which can be addressed via novel task-oriented cross-system designs.

5.2.3 Semantic-Aware, Task-Oriented Communications for the Metaverse

Recently, semantic-aware, task-oriented communications - attracts significant attention from academia and industry, which is expected to break through some key communication challenges by leveraging the advances of deep machine learning. In principle, this technology advocates the paradigm shift from today's communications that deliver information bits (irrespective of the meaning conveyed by the transmitted bits) to future communications that deliver the meaning and complete tasks [187]. It deeply couples communications and its applications, which requires cross-system design. For example, JSCC is one of the typical semantic communication design strategies that jointly train the source and channel coding. It has been shown in [23] that JSCC is superior in reconstructing in-text sentences, sounds, images, and point cloud data. In particular, it can achieve significant performance gains at low signal-to-noise ratio circumstances. However, most of the current JSCC research activities are limited in the areas of data reconstruction and image retrieval (CV) - the tasks that are well-defined and can be achieved by existing neural networks with explicitly derivable objective functions [23]. Other challenges include (1) Large computation cost: Training deep neural networks for JSCC or other types of semantic-aware and task-oriented communications requires a large amount of computational resources, which is not feasible for mobile devices; (2) Generalization: Switching tasks or contexts usually require to train new neural networks, which impedes the wide adoption of this technology for a variety of tasks in the Metaverse. Thus, how to design JSCC and other task-oriented methods for real-world applications in the Metaverse is still an open question.

5.2.4 Wireless Networked Control Systems

The task-oriented cross-system design between communications and distributed control systems - Wireless Networked Control System (WNCS) - is an important topic for the industrial Metaverse. The basic idea is to use the wireless network to connect the essential elements of one or multiple closed-loop control systems, such as sensors, actuators, and controllers, and other HCI devices, including tablet, AR/VR devices [188]. This allows the flexible configuration of industry automation for a variety of tasks, such as remote sensing, monitoring, maintenance, and fast reconfiguration of production line/modules, across different sectors, including intelligent transportation, digital healthcare, Industrial 4.0, power and energy [189].

Sensing, communications, and control in WNCS are deeply coupled and therefore need to be co-designed to achieve the overall system performance [190]. Unlike traditional communications designed primarily for consumable markets that optimize the peak and average system performance while treating the extreme cases as outages, the future telecommunication for industrial markets is significantly different with the focus on handling extreme cases and corner events. Thus, the cross-system design is essential to deliver the overall system performance, such as

reliability, safety, and robustness. These require overcoming many design challenges in 6G, such as modeling, design, optimization, testing, and verification.

5.2.5 URLLC, Tactile Internet, and Internet of Skills

The task-oriented cross-system design that uses domain knowledge outside the communication system is essential to deliver true URLLC services. Currently, URLLC is well recognized as an essential requirement to enable a wide range of real-time applications and emerging concepts, such as tactile Internet or Internet of Skills [180]. The 5G standard has defined URLLC as one of the three core features. Researchers and engineers have made significant efforts aiming at the ultra-low latency and high-reliability requirements, i.e., around 1-millisecond latency and up to 99.99999% reliability, for 5G New Radio [79]. However, there is still no guarantee of the system-level end-to-end performance that includes other factors, such as signal processing delay, computing delay, operating system delay, and the time delay in activating actuators. Thus, the cross-system design is a promising solution to address the URLLC challenge by leveraging the domain knowledge [90] For example, it has been demonstrated in [27] that the high-performance synchronization between a physical robotic arm and its digital counterpart in the Metaverse can be achieved by exploiting domain-knowledge from signal processing and robotics systems.

To bridge the gap between the diverse KPI requirements in the Metaverse and the next-generation URLLC (xURLLC) in 6G, researchers and engineers need to develop novel design methods and innovative technologies. They need to be capable of integrating cross-domain knowledge, including sensing, communications, control, and rendering, to ensure high reliability and low latency for future Metaverse applications.

Appendix A

Proof of Markov Poperties

According to the definition, the k-th state includes, $\mathscr{T}(k-1)$, $\mathscr{T}(k)$, $\hat{\mathscr{T}}(k-1)$ and $\hat{\mathscr{T}}(k)$. As shown in (3.6), the predicted trajectory segments, $\hat{\mathscr{T}}(k-1)$ and $\hat{\mathscr{T}}(k)$, depend on the reconstructed historical trajectory, i.e.,

$$\begin{aligned}&\bar{\mathscr{T}}_{\text{in}}(k)\\&=[\bar{\tau}(t_k-L_{\text{in}}-W_k-W_{k-1}),...,\bar{\tau}(t_k-W_k-W_{k-1})].\end{aligned}$$

Further considering sampling and reconstruction in (3.4) and (3.5), $\bar{\mathscr{T}}_{in}(k)$ is determined by the actions and trajectory from the $(t_k-L_{\text{in}}-W_k-W_{k-1})$-th slot to the $(t_k-W_k-W_{k-1})$-th slot. Therefore, $\mathscr{T}(k-1)$ and $\mathscr{T}(k)$ in the k-th state are determined by the states and actions in the past $(t_k-L_{\text{in}}-W_k-W_{k-1})$ slots, which are available at the transmitter (the agent).

Similarly, the $(k+1)$-th state depends on the states and actions in the past $(t_{k+1}-L_{\text{in}}-W_{k+1}-W_k)$ slots, which are highly overlapped with the states and actions from the $(t_k-L_{\text{in}}-W_k-W_{k-1})$-th slot to the t_k-th slot. The new trajectory information by the end of the $(k+1)$-th state includes $[\tau(t_{k+1}-W_{k+1}),...,\tau(t_{k+1})]$ and the k-th action is $\mathbf{a}_k$. The dimension of the input of the prediction algorithm, L_{in}, is the de-correlation time of the trajectory. Thus, $[\tau(t_{k+1}-W_{k+1}),...,\tau(t_{k+1}))]$ only depends on the trajectory $[\tau(t_k-L_{\text{in}}-W_k-W_{k-1}),...,\tau(t_k)]$ (with the assumption $W_{k+1}<W_k+W_{k-1}$) [1] and action $\mathbf{a}_k$. In Fig. A.1, we illustrate the relationship between de-correlation time and the length of the historical trajectory (observation horizon). Given the k-th state-action pair, if the observation horizon is longer than the de-correlation time of the system, the $(k+1)$-th state does not depend on the states and actions before the $(t_k-L_{\text{in}}-W_k-W_{k-1})$-th slot. According to the definition of the Markov decision process, the Markov property holds in our problem.

[1] This assumption holds in most cases in our system since the prediction horizons of two consecutive segments are highly correlated and do not vary rapidly.

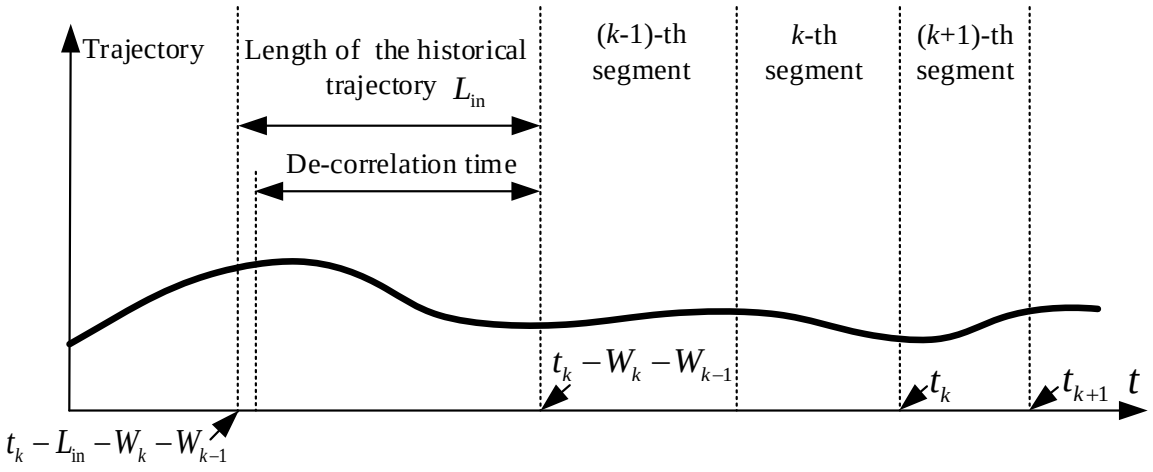


Figure A.1: The relationship between the de-correlation time and length of the historical trajectory.

Appendix B

Calculation of Jacobian Matrix

For notational simplicity, the notations used in the appendix are different from the notations used in the main text.

To obtain the Jacobian matrix defined in Section 4.3.2, one approach is to compute the partial derivatives with respect to each joint angle. The computation complexity of this approach could be high, and we introduce a low-complexity numerical method in this appendix to obtain the Jacobian matrix [155]. For a robotic arm with I rotation joints, the Jacobian matrix can be obtained from the following expression,

$$\mathscr{J} = \begin{bmatrix} {}^{0}_{0}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^{0}_{I}\xi - {}^{0}_{0}\xi) & {}^{0}_{1}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^{0}_{I}\xi - {}^{0}_{1}\xi) & \cdots & {}^{0}_{I-1}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^{0}_{I}\xi - {}^{0}_{I-1}\xi) \\ {}^{0}_{0}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^{0}_{1}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \cdots & {}^{0}_{I-1}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{6\times I}, \tag{B.1}$$

where $\times$ is the cross product operation defined by: $\mathbf{a} = [x_1, y_1, z_1]$, $\mathbf{b} = [x_2, y_2, z_2]$, $\mathbf{a} \times \mathbf{b} = [y_1 z_2 - y_2 z_1,\ x_2 z_1 - x_1 z_2,\ x_1 y_2 - x_2 y_1]^{\mathrm{T}}$, ${}^{0}_{I-1}R \in \mathbb{R}^{3\times 3}$ is the rotation matrix that describes the rotation of the coordinate frame $\{I-1\}$ in the coordinate frame $\{0\}$ which is the base coordinate and ${}^{0}_{I-1}\xi \in \mathbb{R}^{3\times 1}$ is the translation vector that describes the translation of the origin of the coordinate frame $\{I-1\}$ in the coordinate frame $\{0\}$. In robotics, a coordinate frame is a system of reference used to describe the position and motion of robots in space. As shown in Fig. B.1, the rotation center of a joint is commonly used as the reference point for setting up the coordinate frame $\{i\}$. Then, by concatenating the rotation matrix and the translation vector, the

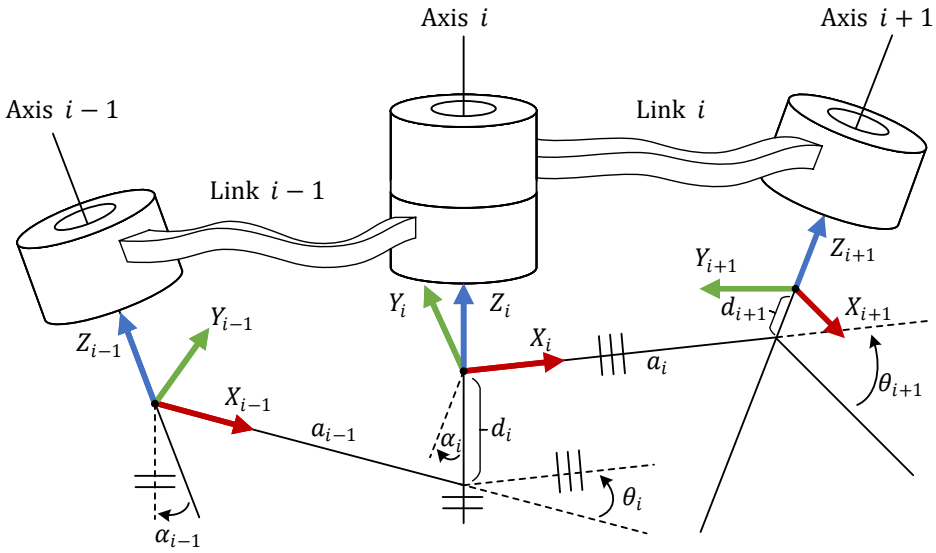


Figure B.1: The coordinate frame $\{i\}$ and D-H parameters.

transformation matrix of ${}^{0}_{I-1}T$ is expressed as

$$ {}^{0}_{I-1}T = \left[\begin{array}{c|c} {}^{0}_{I-1}R & {}^{0}_{I-1}\xi \\ \hline [0,0,0] & 1 \end{array}\right] \in \mathbb{R}^{4\times 4}, \tag{B.2} $$

which describes the relative position and orientation of coordinate frame $\{I-1\}$ with respect to the coordinate frame $\{0\}$.

One way to obtain the transformation matrix is to derive the D-H parameters [155]. D-H parameters provide a systematic way to describe the position and orientation of each link and joint in the robot in the joint space which is widely used by industrial manufacturers. The relationship between D-H parameters and transformation matrix ${}^{i-1}_{i}T$ can be expressed by

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin(\alpha_{i-1})d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.3} $$

As shown in Fig B.1, θ_i is the angle value of i-th joint, α_{i-1}, a_{i-1}, d_i are the constant parameters determined by the mechanical system. Then, the transformation matrix ${}^{0}_{i}T$ between coordinate frames $\{0\}$ and $\{i\}$ can be obtained by the forward kinematic chain [155],

$$ {}^{0}_{i}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\cdots{}^{i-1}_{i}T \tag{B.4} $$

Appendix C

Foundation of Quaternion

Specifically, imaginary basis vectors follow the multiplication rules as

$$\boldsymbol{u}_{\mathrm{i}}^2 = \boldsymbol{u}_{\mathrm{j}}^2 = \boldsymbol{u}_{\mathrm{k}}^2 = -1, \qquad \boldsymbol{u}_{\mathrm{i}}\boldsymbol{u}_{\mathrm{j}}\boldsymbol{u}_{\mathrm{k}} = -1 \tag{C.1}$$

$$\boldsymbol{u}_{\mathrm{i}}\boldsymbol{u}_{\mathrm{j}} = -\boldsymbol{u}_{\mathrm{j}}\boldsymbol{u}_{\mathrm{i}} = \boldsymbol{u}_{\mathrm{k}}, \qquad \boldsymbol{u}_{\mathrm{j}}\boldsymbol{u}_{\mathrm{k}} = -\boldsymbol{u}_{\mathrm{k}}\boldsymbol{u}_{\mathrm{j}} = \boldsymbol{u}_{\mathrm{i}}, \qquad \boldsymbol{u}_{\mathrm{k}}\boldsymbol{u}_{\mathrm{i}} = -\boldsymbol{u}_{\mathrm{i}}\boldsymbol{u}_{\mathrm{k}} = \boldsymbol{u}_{\mathrm{j}}. \tag{C.2}$$

As shown in Fig. 4.3, according to the unit vector of the rotation axis $\boldsymbol{\eta}_{\mathrm{c}}$ and the rotation angle ψ_η, the quaternion can be obtained by

$$\begin{aligned}\boldsymbol{q} &= F_q(a_{\mathrm{c}}, b_{\mathrm{c}}, c_{\mathrm{c}}, \psi_\eta) \\ &= \sin(\frac{\psi_\eta}{2}) \cdot a_{\mathrm{c}} \cdot \boldsymbol{u}_{\mathrm{i}} + \sin(\frac{\psi_\eta}{2}) \cdot b_{\mathrm{c}} \cdot \boldsymbol{u}_{\mathrm{j}} + \sin(\frac{\psi_\eta}{2}) \cdot c_{\mathrm{c}} \cdot \boldsymbol{u}_{\mathrm{k}} + \cos(\frac{\psi_\eta}{2}).\end{aligned} \tag{C.3}$$

Joint	a (m)	d (m)	α (rad)	θ (rad)
Joint 1	0	0.333	0	θ_1
Joint 2	0	0	$-\frac{\pi}{2}$	θ_2
Joint 3	0	0.316	$\frac{\pi}{2}$	θ_3
Joint 4	0.0825	0	$\frac{\pi}{2}$	θ_4
Joint 5	-0.0825	0.384	$-\frac{\pi}{2}$	θ_5
Joint 6	0	0	$\frac{\pi}{2}$	θ_6
Joint 7	0.088	0	$\frac{\pi}{2}$	θ_7

Table C.1: D-H Parameters of Franka Emika Panda Robotic Arm

In our prototype, the Franka Emika Panda robotic arm is used. The corresponding D-H parameters are shown in Table C [191]. Thus, by substituting the D-H parameters into (B.4), we can obtain ${}^0_iT$, ${}^0_iR$ and ${}^0_i\xi$, $i = 1, 2, ..., I$. Then, by substituting ${}^0_iR$ and ${}^0_i\xi$, $i = 1, 2, ..., I$, into (B.1), we can obtain the Jacobian matrix $\mathscr{J}$. This completes the calculation of the Jacobian matrix.

Table 5.2: Summary of Main Symbols

Symbol	Explanation
t_s	Duration of time slot
I	The number of joints
$\mathscr{T}(t)$	Trajectory of the real-world robotic arm in the t-th time slot
$\tau_i(t)$	Angle value of the i-th joint measured in the t-th time slot
$x_i(t)$	Whether the i-th joint is scheduled to transmit in the t-th time slot
$X(t)$	Decision of the scheduler in the t-th time slot
$y_i(t)$	Indicators of whether packet arrivals in the t-th time slot
$\phi(t)$	Angle between x-axis and x'-axis
W_l	Historical observation window for reconstruction
$\bar{\mathscr{T}}(t_0)$	Reconstructed trajectory in the t_0 time slot
$F_l(\cdot,\theta_l)$	Function of reconstruction
θ_l	Parameters of reconstruction
W_p	Input length of the prediction
H	Output length of the prediction
$F_p(\cdot,\theta_p)$	Function of prediction
θ_p	Parameters of the prediction function
L_{p}	MSE loss of prediction
$\hat{\mathscr{T}}(t)$	Prediction trajectory in the t-th time slot
$\hat{\tau}_i(t)$	Prediction trajectory of the i-th joint in the t-th time slot
$Z(t)$	Optimal prediction horizon
$z_i(t)$	Optimal prediction horizon of the i-th joint
N_c	Time interval of control command generation
N_r	Processing time of each image
k_p	Proportional parameter of PD controller
k_d	Derivative parameter of PD controller
$\widetilde{\mathscr{T}}(t)$	Control results in the t-th time slot
$\widetilde{\tau}_i(t)$	Control result of the i-th joint in the t-th time slot
$F_r(\cdot,\theta_r)$	Function of rendering
θ_r	Parameters of rendering function
$\check{\mathscr{T}}(k)$	The rendered trajectory
$\mathscr{P}(t)$	Pose of real-world robotic arm in the t-th time slot
$\boldsymbol{\eta}_c(t)$	Unit vector of rotation axis
$\psi_\eta(t)$	Angle of rotation

Symbol	Explanation
$F_f(\cdot)$	Forward kinematics of real-world robotic
$\check{P}(t)$	Pose of virtual-world robotic arm in the t-th time slot
$\mathbf{e}(t)$	Modeling error in the t-th time slot
ω_1, ω_2	Weighting coefficients of the modeling error
π_θ	Scheduling and prediction policy
θ	Parameters of policy
${a_{t,i}}^{[1]}, {a_{t,i}}^{[2]}$	Action of the i-th joint taken in the t-th time slot
$\dot{\mathbb{S}}(t)$	Set of angles received by the Metaverse
${\rho_{t,i}}^{[1]}, {\rho_{t,i}}^{[2]}$	Distribution of action in the t-th time slot
$\mathscr{L}_c$, $\mathscr{L}_r$	Loss functions of policy network
$A(\mathbf{s}, \mathbf{a})$	Advantage function
$Q^{\pi_\theta}(\mathbf{s}, \mathbf{a})$	State-action value under policy
$V^{\pi_\theta}(\mathbf{s})$	State value function
$\mathscr{J}(\cdot)$	Jacobian matrix
L_1, L_2, L_3	Length of joint links
$r(\mathbf{s}_t, \mathbf{a_t})$	Instantaneous reward
$c(\mathbf{s}_t, \mathbf{a_t})$	Instantaneous cost
$\mathbf{a}_t$	Action taken in the t-th time slot
C^{π_θ}	Long-term cost
β	Learning rate of CVaR
$\mathbf{s}_t$	State
v	Multiplier of CVaR
Γ_c	Constraint of modeling error
R^{π_θ}	Long-term reward
γ	Discount factor
$\mathrm{CVaR}_\alpha(\cdot)$	CVaR function
1-α	Confidence level of CVaR

Bibliography

[1] A. M. Aslam, R. Chaudhary, A. Bhardwaj, I. Budhiraja, N. Kumar, and S. Zeadally, "Metaverse for 6G and beyond: The next revolution and deployment challenges," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 32–39, 2023.

[2] N. Stephenson, *Snow crash: A novel*. Spectra, 2003.

[3] L. U. Khan, M. Guizani, D. Niyato, A. Al-Fuqaha, and M. Debbah, *Metaverse for wireless systems: Architecture, advances, standardization, and open challenges*, 2023. [Online]. Available: https://arxiv.org/abs/2301.11441.

[4] L.-H. Lee, T. Braud, P. Zhou, *et al.*, *All one needs to know about Metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda*, 2021. [Online]. Available: https://arxiv.org/abs/2110.05352.

[5] BMW, *Industry 4.0*, www.bmwgroup.com, 2022. [Online]. Available: https://www.bmwgroup.com/en/innovation/company/industry-4-0.html.

[6] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch, "The effect of avatar realism in immersive social virtual realities," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, 2017, pp. 1–10.

[7] C. Collins, "Looking to the future: Higher education in the Metaverse," *Educause Review*, vol. 43, no. 5, pp. 51–63, 2008.

[8] K. Wang, J. Jin, Y. Yang, *et al.*, *Task offloading with multi-tier computing resources in next generation wireless networks*, 2022. [Online]. Available: https://arxiv.org/abs/2205.13866.

[9] 3GPP, "Study on scenarios and requirements for next generation access technologies," TSG RAN TR38.913 R14, Jun. 2017.

[10] A. M. Girgis, J. Park, M. Bennis, and M. Debbah, "Predictive control and communication co-design via two-way gaussian process regression and AoI-aware scheduling," *IEEE Transaction of Communications*, vol. 69, no. 10, pp. 7077–7093, 2021.

[11] F. Liu, Y. Cui, C. Masouros, *et al.*, "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1728–1767, 2022.

[12] M. Xu, W. C. Ng, W. Y. B. Lim, *et al.*, "A full dive into realizing the edge-enabled Metaverse: Visions, enabling technologies, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 656–700, 2023.

[13] A. M. Girgis, J. Park, C.-F. Liu, and M. Bennis, "Predictive control and communication co-design: A gaussian process regression approach," in *Proceedings of 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2020, pp. 1–5.

[14] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[15] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, 2021, ISSN: 1389-1286. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000773.

[16] C. She, C. Yang, and T. Q. S. Quek, "Cross-layer optimization for ultra-reliable and low-latency radio access networks," *IEEE Transaction of Wireless Communications*, vol. 17, no. 1, pp. 127–141, 2018.

[17] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2015, pp. 3431–3440.

[19] S. Liu, Y. Yu, X. Lian, *et al.*, "Dependent task scheduling and offloading for minimizing deadline violation ratio in mobile edge computing networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 538–554, 2023.

[20] J. Yu, A. Alhilal, P. Hui, and D. H. K. Tsang, *6G mobile-edge empowered Metaverse: Requirements, technologies, challenges and research directions*, 2023.

[21] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the edge: How to deliver 360 videos in mobile networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 30–35.

[22] Q. Lan, D. Wen, Z. Zhang, *et al.*, *What is semantic communication? a view on conveying meaning in the era of machine intelligence*, 2021. [Online]. Available: https://arxiv.org/abs/2110.00196.

[23] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, *Less data, more knowledge: Building next generation semantic communication networks*, 2022. [Online]. Available: https://arxiv.org/abs/2211.14343.

[24] B. Blau, C. E. Hughes, M. J. Moshell, and C. Lisle, "Networked virtual environments," in *Proceedings of the 1992 symposium on Interactive 3D graphics*, 1992, pp. 157–160.

[25] L. d. studio, *Beat saber - VR rhythm game*, beatsaber.com. [Online]. Available: https://beatsaber.com/.

[26] WayRay, *Wayray*, wayray.com, 2022. [Online]. Available: https://wayray.com/.

[27] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in Metaverse," *IEEE Journal on Selected Areas in Communications*, 2022.

[28] F. Hu, Y. Deng, H. Zhou, T. H. Jung, C.-B. Chae, and A. H. Aghvami, "A vision of an XR-aided teleoperation system toward 5G/B5G," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 34–40, 2021.

[29] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[30] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 399–404.

[31] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 15 159–15 168.

[32] G. Carneiro, N. P. Da Silva, A. Del Bue, and J. P. Costeira, "Artistic image classification: An analysis on the printart database," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 10 7-13, 2012, Proceedings, Part IV 12*, Springer, 2012, pp. 143–157.

[33] X. Xu, Y. Gao, K. Yan, X. Lin, and Q. Zou, "Location-free human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 13 137–13 146.

[34] G. Ning, P. Liu, X. Fan, and C. Zhang, "A top-down approach to articulated human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, Sep. 2018.

[35] Z. Yu, B. Ni, J. Xu, J. Wang, C. Zhao, and W. Zhang, "Towards alleviating the modeling ambiguity of unsupervised monocular 3D human pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 8651–8660.

[36] B. Artacho and A. Savakis, "Unipose: Unified human pose estimation in single images and videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[37] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[38] F. Pizzati, P. Cerri, and R. de Charette, "Comogan: Continuous model-guided image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 14 288–14 298.

[39] B. Zhu, Z. Wang, S. Shi, H. Xu, L. Hong, and H. Li, "Conquer: Query contrast voxel-detr for 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 9296–9305.

[40] R. Zhang, T. Pfister, and J. Li, *Harmonic unpaired image-to-image translation*, 2019. [Online]. Available: https://arxiv.org/abs/1902.09727.

[41] A. Gokaslan, V. Ramanujan, D. Ritchie, K. I. Kim, and J. Tompkin, "Improving shape deformation in unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

[42] J. Lin, Y. Xia, T. Qin, Z. Chen, and T.-Y. Liu, "Conditional image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[43] H. Wu, Z. Yao, J. Wang, and M. Long, "Motionrnn: A flexible model for video prediction with spacetime-varying motions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 15 435–15 444.

[44] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[45] B. Wang, T. Hu, B. Li, X. Chen, and Z. Zhang, "Gatector: A unified framework for gaze object prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 19 588–19 597.

[46] Y. Bao, Y. Liu, H. Wang, and F. Lu, "Generalizing gaze estimation with rotation consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 4207–4216.

[47] K. Li, Y. Tang, V. A. Prisacariu, and P. H. Torr, "Bnv-fusion: Dense 3D reconstruction using bi-level neural volume fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 6166–6175.

[48] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3D reconstruction from single and multi-view images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[49] L. Chen, T. Yang, X. Zhang, W. Zhang, and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 8823–8832.

[50] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[51] R. Dey and V. N. Boddeti, "Generating diverse 3D reconstructions from a single occluded face image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 1547–1557.

[52] B. Kim, J. Mun, K.-W. On, M. Shin, J. Lee, and E.-S. Kim, "Mstr: Multi-scale transformer for end-to-end human-object interaction detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 19 578–19 587.

[53] L. Yang, K. Li, X. Zhan, *et al.*, "Oakink: A large-scale knowledge repository for understanding hand-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 20 953–20 962.

[54] Y.-L. Li, X. Liu, H. Lu, *et al.*, "Detailed 2D-3D joint representation for human-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[55] D. Müller, I. Soto-Rey, and F. Kramer, "Towards a guideline for evaluation metrics in medical image segmentation," *BMC Research Notes*, vol. 15, no. 1, pp. 1–8, 2022.

[56] A. T. Kouanou, D. Tchiotsop, R. Tchinda, C. T. Tchapga, A. N. K. Telem, and R. Kengne, "A machine learning algorithm for biomedical images compression using orthogonal transforms," *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. 11, p. 38, 2018.

[57] M. O. Kaplan and S. E. Alptekin, "An improved bigan based approach for anomaly detection," *Procedia Computer Science*, vol. 176, pp. 185–194, 2020.

[58] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 7210–7219.

[59] D. Peng, Y. Lei, M. Hayat, Y. Guo, and W. Li, "Semantic-aware domain generalized segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 2594–2605.

[60] Q. Shuai, C. Geng, Q. Fang, *et al.*, "Novel view synthesis of human interactions from sparse multi-view videos," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.

[61] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 5885–5894.

[62] X. Ma, J. Su, C. Wang, H. Ci, and Y. Wang, "Context modeling in 3D human pose estimation: A unified perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 6238–6247.

[63] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "Focalclick: Towards practical interactive image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 1300–1309.

[64] J. Lee, J. Yi, C. Shin, and S. Yoon, "Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 2643–2652.

[65] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[66] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[67] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[68] X. You, C.-X. Wang, J. Huang, *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, pp. 1–74, 2021.

[69] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020.

[70] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, *Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline*, 2022.

[71] W. Zielonka, T. Bolkart, and J. Thies, "Instant volumetric head avatars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2023, pp. 4574–4584.

[72] K. Li, T. Rolff, S. Schmidt, R. Bacher, W. Leemans, and F. Steinicke, "Interacting with neural radiance fields in immersive virtual reality," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–4.

[73] R. A. Fisher *et al.*, "012: A mathematical examination of the methods of determining the accuracy of an observation by the mean error, and by the mean square error.," 1920.

[74] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[75] M. Andriluka, U. Iqbal, E. Insafutdinov, *et al.*, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[76] J. Gu, F. Xiang, X. Li, *et al.*, *Maniskill2: A unified benchmark for generalizable manipulation skills*, 2023.

[77] B. Osiński, P. Miłoś, A. Jakubowski, *et al.*, *Carla real traffic scenarios–novel training ground and benchmark for autonomous driving*, 2020. [Online]. Available: https://arxiv.org/abs/2012.11329.

[78] Z. Chen, M. Xu, C. She, Y. Jia, M. Wang, and Y. Li, "Improving timeliness-fidelity tradeoff in wireless sensor networks: Waiting for all and waiting for partial sensor nodes," *IEEE Transactions on Communications*, 2023.

[79] C. She, C. Sun, Z. Gu, *et al.*, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 204–246, 2021.

[80] H. Tataria, M. Shafi, M. Dohler, and S. Sun, "Six critical challenges for 6G wireless systems: A summary and some solutions," *IEEE Vehicular Tech. Mag.*, vol. 17, no. 1, pp. 16–26, 2022.

[81] R. Merritt, *What is accelerated computing?* Sep. 2022. [Online]. Available: https://blogs.nvidia.com/blog/2021/09/01/what-is-accelerated-computing/.

[82] T. Müller, *tiny-cuda-nn*, version 1.7, Apr. 2021. [Online]. Available: https://github.com/NVlabs/tiny-cuda-nn.

[83] J. J. Q. Omana, M. M. Marinho, and K. Harada, "Digital twin of a multi-arm robot platform based on isaac sim for synthetic data generation," in *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023.

[84] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[85] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: Lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[86] T. Mu, Z. Ling, F. Xiang, *et al.*, *Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations*, 2021. [Online]. Available: https://arxiv.org/abs/2107.14483.

[87] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[88] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[89] Y. Wang, Z. Su, N. Zhang, *et al.*, "A survey on Metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 319–352, 2023.

[90] C. She, C. Sun, Z. Gu, *et al.*, "A tutorial of ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Procedings of the IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.

[91] NVIDIA, *Omniverse platform for virtual collaboration*, NVIDIA. [Online]. Available: https://www.nvidia.com/en-us/omniverse/.

[92] S. Zhang, W. Y. B. Lim, W. C. Ng, *et al.*, "Towards green Metaverse networking: Technologies, advancements and future directions," *IEEE Network*, pp. 1–10, 2023.

[93] Y. Huang, Y. J. Li, and Z. Cai, "Security and privacy in Metaverse: A comprehensive survey," *Big Data Mining and Analytics*, vol. 6, no. 2, pp. 234–247, 2023.

[94] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[95] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the wiener process for remote estimation over a channel with random delay," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 1118–1135, 2020.

[96] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, 2021.

[97] S. Wang, M. Chen, W. Saad, C. Yin, S. Cui, and H. V. Poor, "Reinforcement learning for minimizing age of information under realistic physical dynamics," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–6.

[98] H. Tang, J. Wang, L. Song, and J. Song, "Minimizing age of information with power constraints: Multi-user opportunistic scheduling in multi-state time-varying channels," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 854–868, 2020.

[99] E. Fountoulakis, M. Codreanu, A. Ephremides, and N. Pappas, *Joint sampling and transmission policies for minimizing cost under aoi constraints*, 2021. [Online]. Available: https://arxiv.org/abs/2103.15450.

[100] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.

[101] N. Pappas and M. Kountouris, "Goal-oriented communication for real-time tracking in autonomous systems," in *Proceedings of IEEE International Conference on Autonomous Systems (ICAS)*, 2021, pp. 1–5.

[102] Y. Sun and B. Cyr, "Information aging through queues: A mutual information perspective," in *Proceedings of IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[103] X. Hou and S. Dey, "Motion prediction and pre-rendering at the edge to enable ultra-low latency mobile 6dof experiences," *IEEE Open Journal on Communication Society*, vol. 1, pp. 1674–1690, 2020.

[104] F. Richter, Y. Zhang, Y. Zhi, R. K. Orosco, and M. C. Yip, "Augmented reality predictive displays to help mitigate the effects of delayed telesurgery," in *Proceedings of IEEE International Conference Robotics and Automation (ICRA)*, 2019, pp. 444–450.

[105] X. Tong, G. Zhao, M. A. Imran, Z. Pang, and Z. Chen, "Minimizing wireless resource consumption for packetized predictive control in real-time cyber physical systems," in *Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[106] Z. Hou, C. She, Y. Li, L. Zhuo, and B. Vucetic, "Prediction and communication co-design for ultra-reliable and low-latency communications," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1196–1209, 2019.

[107] B. Kizilkaya, C. She, G. Zhao, and M. A. Imran, "Task-oriented prediction and communication co-design for haptic communications," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 8987–9001, 2023.

[108] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *Journal of the Indian Institute of Science*, vol. 100, pp. 435–443, 2020.

[109] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[110] F. Tang, X. Chen, M. Zhao, and N. Kato, "The roadmap of communication and networking in 6G for the Metaverse," *IEEE Wireless Communications*, pp. 1–15, 2022.

[111] R. De Gaudenzi, M. Luise, and L. Sanguinetti, "The open challenge of integrating satellites into (beyond-) 5G cellular networks," *IEEE Network*, vol. 36, no. 2, pp. 168–174, 2022.

[112] A. Lacava, M. Polese, R. Sivaraj, *et al.*, "Programmable and customized intelligence for traffic steering in 5G networks using open ran architectures," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2023.

[113] B. Siniarski, C. D. Alwis, G. Yenduri, *et al.*, *Need of 6G for the Metaverse realization*, 2022.

[114] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020.

[115] G. Szabó, S. Rácz, N. Reider, H. A. Munz, and J. Pető, "Digital twin: Network provisioning of mission critical communication in cyber physical production systems," in *Proc. 2019 IEEE Int. Conf. Ind. 4.0 Artif. Intell. Commun. Technol. (IAICT)*, IEEE, 2019, pp. 37–43.

[116] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, 2019.

[117] G. Minopoulos and K. E. Psannis, "Opportunities and challenges of tangible xr applications for 5G networks and beyond," *IEEE Consumer Electronics Magazine*, 2022.

[118] A. Qayyum, M. A. Butt, H. Ali, *et al.*, *Secure and trustworthy artificial intelligence-extended reality (ai-xr) for Metaverses*, 2022. [Online]. Available: https://arxiv.org/abs/2210.13289.

[119] S. K. Jagatheesaperumal, Z. Yang, Q. Yang, *et al.*, *Semantic-aware digital twin for Metaverse: A comprehensive review*, 2023. [Online]. Available: https://arxiv.org/abs/2305.18304.

[120] V.-P. Bui, S. R. Pandey, F. Chiariotti, and P. Popovski, *Game networking and its evolution towards supporting Metaverse through the 6G wireless systems*, 2023. [Online]. Available: https://arxiv.org/abs/2302.01672.

[121] M. Xu, D. Niyato, B. Wright, *et al.*, *EPViSA: Efficient auction design for real-time physical-virtual synchronization in the Metaverse*, 2022. [Online]. Available: https://arxiv.org/abs/2211.06838.

[122] Y. Han, D. Niyato, C. Leung, *et al.*, "A dynamic hierarchical framework for IoT-assisted digital twin synchronization in the Metaverse," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 268–284, 2023.

[123] S. K. Jagatheesaperumal, K. Ahmad, A. Al-Fuqaha, and J. Qadir, *Advancing education through extended reality and internet of everything enabled Metaverses: Applications, challenges, and open issues*, 2022. [Online]. Available: https://arxiv.org/abs/2207.01512.

[124] N. H. Chu, D. N. Nguyen, D. T. Hoang, *et al.*, *Dynamic resource allocation for Metaverse applications with deep reinforcement learning*, 2023. [Online]. Available: https://arxiv.org/abs/2302.13445.

[125] O. Hashash, C. Chaccour, W. Saad, K. Sakaguchi, and T. Yu, *Towards a decentralized Metaverse: Synchronized orchestration of digital twins and sub-Metaverses*, 2022. [Online]. Available: https://arxiv.org/abs/2211.14686.

[126] S. Zeng, Z. Li, H. Yu, *et al.*, *HFedMS: Heterogeneous federated learning with memorable data semantics in industrial Metaverse*, 2022. [Online]. Available: https://arxiv.org/abs/2211.03300.

[127] P. M. d. Santana, N. Marchenko, B. Soret, and P. Popovski, "Goal-oriented wireless communication for a remotely controlled autonomous guided vehicle," *IEEE Wireless Communications Letter*, 2023.

[128] D. Wen, P. Liu, G. Zhu, *et al.*, *Task-oriented sensing, computation, and communication integration for multi-device edge AI*, 2022. [Online]. Available: https://arxiv.org/abs/2207.00969.

[129] W. Yu, T. J. Chua, and J. Zhao, *User-centric heterogeneous-action deep reinforcement learning for virtual reality in the Metaverse over wireless networks*, 2023. [Online]. Available: https://arxiv.org/abs/2302.01471.

[130] 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.

[131] Rajratnakharat, V. Bavane, S. Jadhao, and R. Marode, "Digital twin: Manufacturing excellence through virtual factory replication," in *Proceedings of Recent Advances in Science and Engineering*, 2018.

[132] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. air force vehicles," in *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Struction, Structural Dynamics*, 2012, pp. 1–14.

[133] "The Cyber-Physical Infrastructure - Empowering innovation, people, robots and smart machines to enhance prosperity, resilience, sustainability and security," en, p. 65,

[134] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 219–16 230, 2021.

[135] O. Hashash, C. Chaccour, and W. Saad, *Edge continual learning for dynamic digital twins over wireless networks*, 2022.

[136] V.-P. Bui, S. R. Pandey, F. Chiariotti, and P. Popovski, "Scheduling policy for Value-of-Information (VoI) in trajectory estimation for digital twins," *IEEE Communications Letter*, 2023.

[137] I. A. Tsokalo, D. Kuss, I. Kharabet, F. H. Fitzek, and M. Reisslein, "Remote robot control with human-in-the-loop over long distances using digital twins," in *Proceedings of 2019 IEEE Global Communication Conference (GLOBECOM)*, IEEE, 2019, pp. 1–6.

[138] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industry Information*, vol. 18, no. 2, pp. 1367–1376, 2020.

[139] 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. 20 325–20 336, 2019.

[140] D. Holloway, "Virtual worlds and health: Healthcare delivery and simulation opportunities," in *Virtual worlds and Metaverse platforms: New communication and identity paradigms*, IGI Global, 2012, pp. 251–270.

[141] S. Bardzell and K. Shankar, "Video game technologies and virtual design: A study of virtual design teams in a Metaverse," in *Proceedings of International Conference on Virtual Reality*, Springer, 2007, pp. 607–616.

[142] C. Choi, J. Jun, J. Heo, and K. Kim, "Effects of virtual-avatar motion-synchrony levels on full-body interaction," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 701–708.

[143] S. Ellis, F. Breant, B. Manges, R. Jacoby, and B. Adelstein, "Factors influencing operator interaction with virtual objects viewed via head-mounted see-through displays: Viewing conditions and rendering latency," in *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality*, 1997, pp. 138–145.

[144] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill, "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity," in *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 2004, pp. 39–47.

[145] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile internet: Haptic communications over next generation 5G cellular networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 82–89, 2017.

[146] I. Oliver, A. Miller, and C. Allison, "Mongoose: Throughput redistributing virtual world," in *Proceedings of the 21st IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–9.

[147] M. Dianatfar, J. Latokartano, and M. Lanz, "Review on existing VR/AR solutions in human-robot collaboration," *Procedia CIRP*, vol. 97, pp. 407–411, 2021.

[148] Z. Hou, C. She, Y. Li, D. Niyato, M. Dohler, and B. Vucetic, "Intelligent communications for tactile internet in 6G: Requirements, technologies, and challenges," *IEEE Communications Magazine*, vol. 59, no. 12, pp. 82–88, 2021.

[149] J. F. Steffensen, *Interpolation*. Courier Corporation, 2006.

[150] G. Scaglia, A. Rosales, L. Quintero, V. Mut, and R. Agarwal, "A linear-interpolation-based controller design for trajectory tracking of mobile robots," *Control Engineering Practice*, vol. 18, no. 3, pp. 318–329, 2010.

[151] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[152] Q. Liang, F. Que, and E. Modiano, "Accelerated primal-dual policy optimization for safe reinforcement learning," *arXiv preprint arXiv:1802.06480*, 2018.

[153] F. E. GmbH. "Requirements on Communication Interface for Panda Research Robot." (Dec 2020). (), [Online]. Available: https://frankaemika.github.io/docs/requirements.html.

[154] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[155] J. Craighead, J. Burke, and R. Murphy, "Using the unity game engine to develop sarge: A case study," in *Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008)*, vol. 4552, 2008.

[156] I. D. O. NaturalPoint, *Optitrack prime 13*, https://www.optitrack.com/cameras/primex-13.html, (accessed: Feb. 10, 2022).

[157] L. Huang, Z. Meng, Z. Deng, C. Wang, L. Li, and G. Zhao, "Toward verifying the user of motion-controlled robotic arm systems via the robot behavior," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 422–22 433, 2021.

[158] E. Fountoulakis, M. Codreanu, A. Ephremides, and N. Pappas, *Joint sampling and transmission policies for minimizing cost under AoI constraints*, 2021. [Online]. Available: https://arxiv.org/abs/2103.15450.

[159] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347.

[160] T. Xu, Y. Liang, and G. Lan, "CRPO: A new approach for safe reinforcement learning with convergence guarantee," in *Proceedings of International Conference on Machine Learning (ICML)*, 2021, pp. 11 480–11 491.

[161] Z. Gu, C. She, W. Hardjawana, *et al.*, "Knowledge-assisted deep reinforcement learning in 5G scheduler design: From theoretical framework to implementation," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2014–2028, 2021.

[162] H. Zhou, S. Zhang, J. Peng, *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceeding of AAAI Conference Artificial Intelligence (AAAI)*, vol. 35, 2021, pp. 11 106–11 115.

[163] S. M. Ross, *Stochastic processes*. John Wiley & Sons, 1995.

[164] J. J. Craig, *Introduction to robotics*. Pearson Education, 2006.

[165] *Motion Generation*, https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/ext_omni_isaac_motion_generation.html#isaac-sim-motion-generation, (accessed Mar. 2023).

[166] A. Tewari, J. Thies, B. Mildenhall, *et al.*, *Advances in neural rendering*, 2021. [Online]. Available: https://arxiv.org/abs/2111.05849.

[167] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton Univ. Press, 1999.

[168] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, Jan. 2006.

[169] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017.

[170] N. Heess, T. B. Dhruva, S. Sriram, J. Lemmon, and D. Silver, *Emergence of locomotion behaviours in rich environments*, 2017.

[171] *Proximal Policy Optimization*, https://spinningup.openai.com/en/latest/algorithms/ppo.html, (accessed Jan. 2020).

[172] *torch.multinomial - Pytorch 2.0*, https://pytorch.org/docs/stable/generated/torch.multinomial.html, (accessed Jul. 2023).

[173] S. Feyzabadi, S. Straube, M. Folgheraiter, E. A. Kirchner, S. K. Kim, and J. C. Albiez, "Human force discrimination during active arm motion for force feedback design," *IEEE Transactions on Haptics*, vol. 6, no. 3, pp. 309–319, 2013.

[174] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, *Risk-constrained reinforcement learning with percentile risk criteria*, 2015. [Online]. Available: https://arxiv.org/abs/1512.01629.

[175] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal on risk*, vol. 2, pp. 21–42, 2000.

[176] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal on Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html.

[177] I. D. O. NaturalPoint, *OptiTrack Prime 13*, https://www.optitrack.com/cameras/primex-13.html, (accessed: Jan. 10, 2023).

[178] *Isaac Gym*, https://developer.nvidia.com/isaac-gym, (accessed Jul. 2022).

[179] Z. Hou, C. She, Y. Li, L. Zhuo, and B. Vucetic, "Prediction and communication co-design for ultra-reliable and low-latency communications," *IEEE Transaction on Wireless Communications*, vol. 19, no. 2, pp. 1196–1209, 2020.

[180] Z. Hou, C. She, Y. Li, D. Niyato, M. Dohler, and B. Vucetic, "Intelligent communications for tactile internet in 6G: Requirements, technologies, and challenges," *IEEE Communications Magzine*, vol. 59, no. 12, pp. 82–88, 2021.

[181] C.-C. Tran and C.-Y. Lin, "An intelligent path planning of welding robot based on multisensor interaction," *IEEE Sensors Journal*, vol. 23, no. 8, pp. 8591–8604, 2023.

[182] R. Zhang, J. Chen, Z. Wang, *et al.*, "A step towards conditional autonomy - robotic appendectomy," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2429–2436, 2023.

[183] N. Correll, K. E. Bekris, D. Berenson, *et al.*, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[184] D. Van Huynh, V.-D. Nguyen, S. Chatzinotas, S. R. Khosravirad, H. V. Poor, and T. Q. Duong, "Joint communication and computation offloading for ultra-reliable and low-latency with multi-tier computing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 521–537, 2023.

[185] X. Li, Y. Cui, J. A. Zhang, F. Liu, D. Zhang, and L. Hanzo, *Integrated human activity sensing and communications*, 2022.

[186] J. Yu, A. Alhilal, P. Hui, and D. H. Tsang, *6G mobile-edge empowered Metaverse: Requirements, technologies, challenges and research directions*, 2022. [Online]. Available: https://arxiv.org/abs/2211.04854.

[187] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, 2021, ISSN: 1389-1286. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000773.

[188] A. M. Girgis, H. Seo, J. Park, M. Bennis, and J. Choi, "Predictive closed-loop remote control over wireless two-way split koopman autoencoder," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 285–23 301, 2022.

[189] P. Popovski, J. J. Nielsen, C. Stefanovic, *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, 2018.

[190] W. Liu, G. Nair, Y. Li, D. Nesic, B. Vucetic, and H. V. Poor, "On the latency, rate, and reliability tradeoff in wireless networked control systems for iiot," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 723–733, 2021.

[191] F. E. GmbH, *Robot and Interface Specifications*, https://frankaemika.github.io/docs/control_parameters.html, (accessed Mar. 21, 2023).