UNIVERSITY OF LEEDS

Cooperation and Coordination in Decentralised Digital Twin Networks

Zhengchang Hua

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds

Faculty of Engineering and Physical Sciences

School of Computer Science

December 2025

Intellectual Property Statement

I confirm that the work submitted is my own, except where work which has formed part of jointly authored publications has been included. My contribution and that of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some of the results and work presented in this thesis have been published in the following papers:

- Chapter 4 and Chapter 5 include material published in:

    1. Zhengchang Hua, Karim Djemame, Nikos Tziritas, and Georgios Theodoropoulos. "A Framework for Digital Twin Collaboration". In: *2024 Winter Simulation Conference (WSC)*. 2024.

- Chapter 6 includes material published in:

    1. Zhengchang Hua, Panagiotis Oikonomou, Karim Djemame, Nikos Tziritas, and Georgios Theodoropoulos. "A Digital Twin-based Multi-Agent Reinforcement Learning Framework for Vehicle-to-Grid Coordination". In: *The 25th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2025)*. 2025.

The above publications are primarily the work of myself. The co-authors on these papers helped in an advisory role along with proofreading publications. The development, experimentation and analysis

were performed entirely by myself.

Acknowledgements

I would like to express my sincerest gratitude to my supervisors, Professor Georgios Theodoropoulos and Professor Karim Djemame, for their continuous support and guidance throughout my PhD journey. Their invaluable insights and encouragement have been instrumental in shaping my research and academic growth. I am particularly grateful for their patience and mentorship in steering me back on track whenever I faced challenges.

I would also like to thank Professor Leandro Soares Indrusiak and Dr Rami Bahsoon for serving as my examiners. I deeply appreciate the time they have dedicated to reviewing this thesis; their expertise and feedback will undoubtedly contribute to the refinement of my work.

My sincere thanks go to Dr Nikos Tziritas for his constructive feedback and suggestions, which have significantly enhanced the quality of this thesis. I would also like to extend my appreciation to Professor Evangelos Pournaras and Professor Zheng Wang for their valuable feedback and suggestions during my transfer and annual reviews. Their insights were crucial in refining my research direction at critical stages.

I am grateful to Dr Panagiotis Oikonomou and Dr Masatoshi Hanai for our collaboration and insightful discussions, which have greatly enriched my research experience.

I would like to thank all my colleagues and friends at the University of Leeds and SUSTech, whose camaraderie and support have made this journey enjoyable and fulfilling. A special thanks to Dr Nan

Zhang, Christian Vergara, Georgios Diamantopoulos, and Jingran Shen - our casual discussions have always been a source of inspiration and motivation. I also wish to thank Yaru Zhao for her assistance with administrative tasks at SUSTech, which was greatly appreciated. I would also like to thank Dr Abdullah Aljulayfi, Dr Fatemeh Banaie, Reem Alqahtani, Latifah Alsalem, Maha Alruwaili, and Abdulaziz Alhindi for their contributions to our reading group sessions. These discussions enriched my perspective and broadened my understanding of various research topics.

I must express my deepest gratitude to my parents for their unwavering support and encouragement throughout my academic journey. Their belief in me has been a constant source of strength.

Finally, I would like to thank Shuyi Chen for her love and support, which has been a pillar of strength during my PhD studies.

Declaration of Generative AI Use

In accordance with university policy, I acknowledge the use of the following tools in the preparation of this thesis:

- **Gemini 3.0** (Google, `https://gemini.google.com/`): Used for proofreading, grammatical refinement, and simulating academic inspections to critique argument clarity.
- **NotebookLM** (Google, `https://notebooklm.google.com/`): Used to assist in the organisation and synthesis of literature sources.

Abstract

The increasing complexity of modern cyber-physical systems, such as smart power grids, necessitates new governance paradigms. Digital Twins, which are live virtual models of physical systems, offer powerful simulation and decision support. However, traditional centralised Digital Twin architectures are ill-suited for large, multi-owner systems. They suffer from scalability bottlenecks, single points of failure, and critical privacy vulnerabilities, as stakeholders are often unwilling to share sensitive raw data with a central authority. This thesis addresses these limitations by delivering a novel architectural framework for decentralised collaboration and an algorithm for intelligent coordination.

The framework enables a federation of autonomous Digital Twins to collaborate by exchanging high-level, privacy-preserving "decision information", such as future intentions or predictions, rather than sensitive raw data. This is facilitated by a distributed middleware that integrates with independent Digital Twin instances. To ensure robustness in this open environment, the framework incorporates a trust mechanism that continuously verifies the historical accuracy of each peer's predictions against real-world outcomes, allowing agents to dynamically weigh the reliability of incoming information.

To enable intelligent, optimised coordination, this thesis further contributes the Digital Twin Assisted Multi-Agent Deep Deterministic Policy Gradient algorithm. This Multi-Agent Reinforcement Learning method introduces a "simulation-assisted critic" that leverages the shared decision-level information to construct a global simulation model for training. This unique approach achieves the global awareness needed for centralised training without requiring access to private agent data, thus

resolving the privacy-performance paradox. The complete framework was validated in a complex Vehicle-to-Grid coordination scenario, where it learned sophisticated control policies and achieved performance comparable to fully centralised, non-private learning algorithms.

Contents

List of Figures

List of Tables

List of Abbreviations

CPS	Cyber-Physical System
CTDE	Centralised Training with Decentralised Execution
Dec-POMDP	Decentralised Partially Observable Markov Decision Process
DER	Distributed Energy Resource
DT	Digital Twin
DT-MADDPG	Digital Twin Assisted Multi-Agent Deep Deterministic Policy Gradient
DTN	Digital Twin Network
ECC	Elliptic Curve Cryptography
EV	Electric Vehicle
FL	Federated Learning
GPS	Global Positioning System
HIL	Hardware-in-the-Loop
ICT	Information and Communication Technology
IoT	Internet of Things

LAN	Local Area Network
LCS	Large Complex System
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MARL	Multi-Agent Reinforcement Learning
MAS	Multi-Agent System
MDP	Markov Decision Process
MPC	Multi-Party Computation
MVI	Minimum Viable Information
NAT	Network Address Translation
P2P	Peer-to-Peer
PKI	Public Key Infrastructure
RL	Reinforcement Learning
SoC	State of Charge
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
V2G	Vehicle-to-Grid
VDN	Value-Decomposition Network
VPP	Virtual Power Plant
WAN	Wide Area Network

Chapter 1

Introduction

This chapter introduces the research context, problem statement, and objectives of this thesis. It begins by outlining the motivation for decentralised Digital Twin (DT) networks in critical infrastructure. It then defines the core research problem: the conflict between privacy and coordination, and formulates the specific research questions and objectives addressed in this work. Finally, it outlines the methodology, contributions, and the structure of the remainder of the thesis.

1.1 Background and Motivation

1.1.1 The Era of Large Complex Systems

In recent years, the modelling and governance of Large Complex Systems (LCSs) have become a focal point of technological research. These systems, which include critical infrastructure such as smart cities, regional logistics networks, and modern energy grids, are defined by their intricate compositions of numerous interacting parts [134].

A prime example of an LCS is the modern power grid, which is currently undergoing a significant

transition toward renewable energy sources. This shift has introduced substantial challenges in managing energy generation and storage. V2G technology has emerged as a vital solution, where Electric Vehicles (EVs) function not just as consumers, but as mobile, aggregated distributed energy storage units [118]. By discharging energy back into the grid during peak demand and charging during valleys, V2G enhances grid stability and promotes renewable energy usage [76]. However, the sheer scale of such systems, often involving thousands of autonomous agents with uncertain mobility patterns, renders traditional, static management techniques insufficient. The overall behavior of these systems cannot be easily predicted by understanding individual parts in isolation, making them too complex for conventional centralised control [112, 61].

1.1.2 The Rise of Digital Twins

To address the management of LCS, Digital Twins have garnered significant attention as a transformative enabling technology for "Cyber-Physical" integration [125, 4]. A Digital Twin is more than a mere digital representation; it is a "live" model of a physical entity that is continuously synchronised with real-time data [37, 52].

This continuous synchronisation provides a powerful promise: the ability to perform real-time monitoring and sophisticated "what-if" simulations. By mirroring the dynamic changes of their physical counterparts, DTs allow stakeholders to simulate potential outcomes before they are applied to the physical system, a level of foresight previously unattainable in complex infrastructure management [18]. However, realising this potential in large, multi-owner systems introduces significant architectural challenges, particularly regarding how these twins are deployed and interconnected.

1.1.3 The Deployment Reality: Centralisation versus Practicality

The prevailing approach to this challenge has been a centralised, cloud-centric architecture. However, constructing a single, monolithic DT for an entire LCS, such as a national V2G network, presents

critical bottlenecks. The simulation-centric nature of DTs requires immense computational power and high synchronisation overhead to maintain an accurate digital representation of thousands of entities in real-time [138, 82].

More importantly, in multi-stakeholder environments, centralisation is often blocked by privacy and security concerns. Individual entities (such as private EV owners or regional power suppliers) are often owned by different stakeholders who are hesitant to share raw, granular data with a central authority due to competitive or privacy risks [136]. This challenge has driven the search for methods that enable collaborative intelligence without data centralisation. Federated Learning has emerged as a leading paradigm in this area, allowing multiple parties to collaboratively train a machine learning model without sharing their underlying private data [83, 148]. However, while FL provides a powerful solution for privacy-preserving *statistical learning*, it is fundamentally misaligned with the core value proposition of a Digital Twin. FL typically produces opaque, "black-box" models that excel at prediction but lack the mechanistic, high-fidelity simulation capabilities required for the "what-if" analysis that defines a true Digital Twin. This critical mismatch: the inability of standard FL to create a network of transparent, simulatable twins, has spurred a shift in focus toward a different collaborative model: the "Federation of Twins," which seeks to enable coordination while preserving the unique simulation-centric nature of each DT [130, 132].

However, this federated model, while providing a structural template for decentralisation, currently lacks a mechanism for autonomous, intelligent coordination. Existing approaches focus on data-sharing or joint model training, leaving a void where system-wide decision-making should be. This creates a deadlock: system operators can either protect stakeholder privacy or achieve global optimisation, but not both.

This thesis resolves this deadlock by introducing a novel framework that enables intelligent coordination without compromising privacy. The core innovation is a paradigm shift from sharing sensitive **raw data** to exchanging high-level, privacy-preserving **decisions**. By combining this "decision-level"

Table 1.1: Comparative Analysis of Existing Paradigms vs. Proposed Framework.

Approach	**Privacy**	**Explainability**	**Scalability**	**Coordination**
Centralised RL	★☆☆	★☆☆	★☆☆	★★★
Federated Learning	★★★	★☆☆	★★☆	★★☆
Blockchain/P2P	★★★	★★☆	★☆☆	★☆☆
Proposed	★★★	★★★	★★★	★★★

communication with a trust verification mechanism and a new simulation-assisted multi-agent learning algorithm, our approach allows autonomous Digital Twins to learn globally optimal strategies. This resolves the conflict between privacy and performance, which leads us to the core challenge of this thesis. Table 1.1 provides a comparative summary of how the proposed framework addresses the limitations of existing paradigms (centralised Reinforcement Learning (RL), federated learning and blockchain/Peer-to-Peer (P2P)) across key dimensions such as privacy, explainability, scalability, and coordination.

1.2 Problem Statement: The Privacy-Coordination Paradox

The limitations of centralised architectures discussed in the previous section necessitate a paradigm shift toward Decentralised Digital Twin architectures. In this model, processing, simulation, and decision-making are distributed to the "edge" of the network, closer to the physical entities themselves. However, this transition reveals a fundamental conflict that serves as the core motivation for this research, defined here as the Privacy-Coordination Paradox.

1.2.1 The Willingness to Cooperate

The success of an LCS, particularly those involving public participation like V2G networks, depends entirely on the participants' "willingness to cooperate" [127]. In multi-stakeholder environments, agents, whether they are private EV owners or independent energy providers, will only join and remain active in the network if their privacy and data sovereignty are guaranteed. This requirement

forces a move away from cloud-centric models toward decentralised edge computing, where sensitive behavioral data remains local to the owner's infrastructure.

1.2.2 The Need for Coordination

Conversely, the primary purpose of managing an LCS is to achieve system-wide goals, such as maintaining grid stability or optimising traffic flow. These goals require a degree of synchronisation and collective intelligence. When systems are decentralised to protect privacy, they naturally create "Information Silos" [129]. Without a "central brain" to oversee the entire state of the system, individual DTs lack the global visibility needed to understand how their local actions impact the broader network.

1.2.3 The Core Conflict

The Privacy-Coordination Paradox emerges from these two competing requirements:

> **The Privacy-Coordination Paradox:** In a multi-owner Large Complex System, the requirement for *Data Sovereignty* (local privacy) is fundamentally at odds with the requirement for *Global Optimisation* (coordinated actions).

This paradox creates a significant deadlock in system design, which we categorise into two failure modes:

- **Privacy-Preserving Isolation:** Decentralisation protects privacy but creates "information silos." Without global context, agents cannot align their actions, leading to system-wide inefficiency or failure (e.g., localised EV charging causing a grid surge).
- **Optimisation-Driven Exposure:** Centralisation enables perfect coordination but requires agents to surrender sensitive data (e.g., mobility patterns), violating the trust necessary for multi-stakeholder participation.

To illustrate this paradox continuously throughout this thesis, consider a real-world V2G coordination scenario. Imagine a local community grid servicing hundreds of EVs. The grid operator aims to maximise the utilisation of local renewable energy and minimise power variance from non-renewable fossil fuels (global optimisation). Simultaneously, each EV owner has strictly personal goals: ensuring their vehicle reaches a target State of Charge (SoC) before their morning departure and maximising their economic revenue by buying energy during off-peak "valley" hours and selling it back during peak pricing periods (local utility).

If this system is managed via **Optimisation-Driven Exposure** (a centralised approach), a master controller or centralized training algorithm would need real-time access to every user's private state: their current battery levels, exact departure schedules, and daily travel requirements. While this allows the central agent to optimally schedule charging and discharging to balance the grid, it requires owners to surrender highly sensitive lifestyle data, representing a massive breach of data sovereignty.

Conversely, if the system defaults to **Privacy-Preserving Isolation** (a fully decentralised approach without coordination), owners might simply program their vehicles to charge independently the moment electricity prices drop at night. This uncoordinated self-interest leads to simultaneous mass-charging, creating severe new demand spikes that can overload local transformers and destabilise the grid.

This motivating example highlights the necessity for a new paradigm: one where autonomous EV agents can learn to balance their personal charging needs and economic goals with the broader grid's stability, all without exposing the raw, private parameters that govern their behaviour.

Existing solutions only partially address this tension. As noted, Federated Learning (FL) enables collaborative model training but often results in opaque "black box" models that lack the explainability and "what-if" simulation capabilities required for critical infrastructure. Similarly, Blockchain technologies can establish trust and record transactions, but they often introduce computational overheads that are unsuitable for the high-frequency control loops required in real-time energy management.

Consequently, there remains a critical gap in current research: the lack of a framework that allows autonomous DTs to learn and execute intelligent, coordinated strategies without the need to expose raw, sensitive data.

1.2.4 Characterisation of the Problem Space

As illustrated in Figure 1.1, this paradox manifests across three critical dimensions that define the problem space and the scope of the proposed solutions:

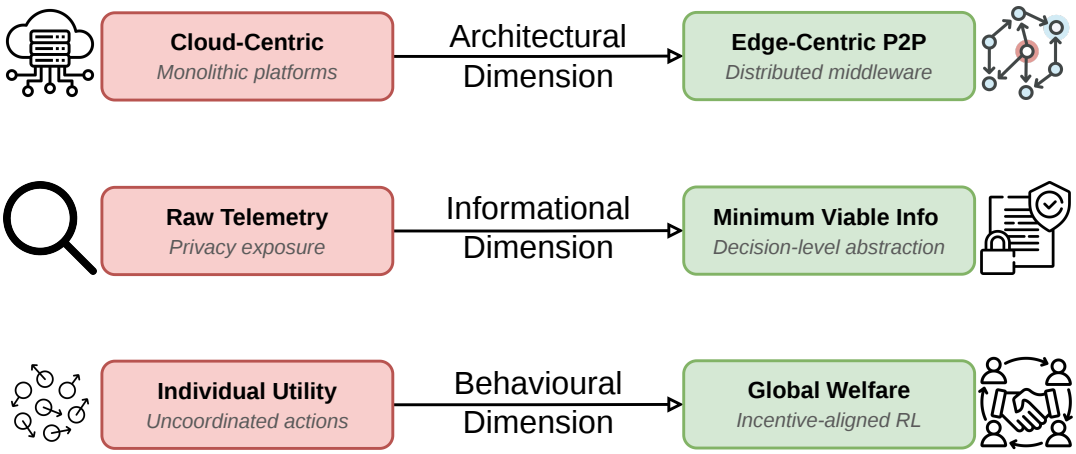

Figure 1.1: The three dimensions of the Privacy-Coordination Paradox illustrating the necessary architectural, informational, and behavioural transitions.

The Architectural Dimension: Involves the structural shift from cloud-centric monolithic platforms to edge-centric, peer-to-peer Digital Twin Networks (DTNs). The challenge here lies in replacing centralised command structures with robust, distributed middleware capable of managing the *Communication Latency*, dynamic topologies, and *Asynchronous State Updates* inherent in decentralised P2P infrastructures. *For example, rather than EVs awaiting charging schedules from a distant central cloud (which is vulnerable to latency and server outages), the architecture must enable EVs to negotiate directly with neighbouring vehicles and local infrastructure.*

The Informational Dimension: Focuses on the fundamental trade-off between *Signal Richness* for

system simulation and strict *Data Privacy*. This dimension addresses what information can safely leave the physical counterpart. It involves determining the Minimum Viable Information (Minimum Viable Information (MVI)), specifically, shifting from raw sensor telemetry to abstracted decision-level data required for agents to coordinate without exposing private state variables. *For example, instead of broadcasting exact GPS coordinates and private travel schedules, an EV shares only a high-level intent, such as "I plan to draw 5kW over the next hour."*

The Behavioural Dimension: Concerns the socio-technical transition from *Individual Utility* to *Global Welfare*. In a multi-owner LCS, agents are often self-interested and autonomous; the problem space requires intelligent learning mechanisms and trust structures that align local incentives with global system stability. *For example, while an EV owner may prefer to charge immediately at maximum speed (individual utility), the system must intelligently incentivise delaying the charge to prevent local transformer overloads (global welfare).*

1.3 Research Question and Objectives

This characterisation of the problem across its architectural, informational, and behavioural dimensions leads directly to the primary research question driving this thesis:

How can autonomous Digital Twins in a decentralised, multi-owner network achieve intelligent, reliable coordination without exposing sensitive raw data?

To answer this question, this thesis pursues four specific research objectives (ROs), each designed to address one or more of the identified dimensions:

- **RO1: Design a Decentralised "Decision-Level" Framework (The Architectural Dimension).** To contribute a novel DT middleware architecture that enables collaboration through the exchange of high-level intentions. This research focuses on establishing a "Decision-Level"

interoperability layer that decouples local simulation from external coordination, ensuring data sovereignty while enabling system-wide visibility.

- **RO2: Develop a Trust Mechanism for Open Systems (The Informational Dimension).** To contribute a mathematical verification protocol that establishes trust based on physical consistency. This research formulates a mechanism to evaluate the veracity of peer predictions against observed reality, enabling robust collaboration in open, untrusted environments without reliance on a central authority.
- **RO3: Propose a Privacy-Preserving MARL Algorithm (The Behavioural Dimension).** To contribute the **Digital Twin Assisted Multi-Agent Deep Deterministic Policy Gradient (DT-MADDPG)** algorithm, a novel hybrid learning method. This research introduces the "Simulation-Assisted Critic" to resolve the conflict between the data requirements of Centralised Training with Decentralised Execution (CTDE) and the privacy constraints of decentralised networks, allowing agents to learn globally optimal strategies.
- **RO4: Validate via Large-Scale V2G Simulation.** To contribute an empirical evaluation of the proposed framework within a realistic Smart Grid scenario. This research quantifies the performance trade-offs between privacy, grid stability, and scalability, demonstrating the efficacy and practicality of the proposed solutions.

1.4 Methodology and Scope

This thesis adopts a **Constructive Research** methodology, a process focused on the design, implementation, and rigorous validation of a novel technical artifact - the Collaborative Digital Twin Framework. By identifying the real-world gap between the need for grid stability and the mandate for user privacy, this research develops a solution that is both theoretically grounded and practically applicable to modern infrastructure challenges.

1.4.1 Scope and Definitions

To ensure conceptual clarity throughout this work, we explicitly distinguish between two fundamental concepts that are often used interchangeably in the literature, yet represent distinct layers of our framework:

- **Cooperation:** Refers to the **infrastructure** and the **willingness** of agents to participate in the network. It encompasses the communication protocols, data sovereignty protections, and trust mechanisms that enable independent DTs to exchange information for the common good. This layer is primarily addressed in Chapters 4 and 5.
- **Coordination and Alignment:** Refers to the **intelligence** and the **alignment** of the system. While cooperation provides the communication channel, coordination involves the algorithmic logic required to achieve alignment. This framework specifically addresses three types of alignment: *Semantic Alignment* (translating heterogeneous internal models into a common language of "intent"), *Temporal Alignment* (synchronising future physical actions to avoid real-world conflicts), and *Objective Alignment* (mathematically reconciling an individual agent's private utility with the global system welfare). This layer is the focus of our Reinforcement Learning developments in Chapter 6.
- **Digital Twinning Scope:** Defines the explicit boundaries and capabilities of the digital twins developed in this research. The *physical counterparts* primarily consist of distributed energy resources and mobile assets, specifically EVs and grid infrastructure components. The *processes* modelled include real-time power consumption, energy storage dynamics, and mobility schedules. Key *features* of these twins include local, high-fidelity physics-based simulation (predictive capabilities) and autonomous decision-making algorithms (prescriptive capabilities). Finally, the *APIs* and network interfaces are strictly constrained to exchange abstracted, decision-level information (e.g., predicted power profiles) rather than raw sensor telemetry,

ensuring interoperability across heterogeneous assets while mathematically guaranteeing data sovereignty.

1.4.2 Validation Domain: Vehicle-to-Grid

While the proposed framework is designed to be applicable to various Large Complex Systems, this research is validated specifically within the domain of **Vehicle-to-Grid** systems. This domain is selected because it represents a "perfect storm" of the challenges identified in the Privacy-Coordination Paradox:

- **High Volatility:** The integration of intermittent renewable energy sources requires extremely responsive and adaptive management.
- **Extreme Decentralisation:** A typical network involves thousands of heterogeneous agents (EVs) operating as autonomous mobile energy pools.
- **Strict Privacy Sensitivity:** EV data includes highly sensitive information regarding owner mobility patterns, home locations, and daily habits, making a centralised "master twin" socially and legally unacceptable.

By solving the coordination problem in the high-stakes environment of V2G, this thesis demonstrates the scalability and reliability of the proposed approach for the broader era of smart, decentralised infrastructure.

1.5 Main Contributions

This thesis makes three primary contributions to the fields of Distributed Systems and Multi-Agent Learning:

1. **The Collaborative Digital Twin Framework:** A novel peer-to-peer middleware that decou-

ples internal state execution from external information sharing. By standardizing "Decision-Level" interaction, it enables heterogeneous agents to cooperate while maintaining strict data sovereignty.

2. **The Trust-Based Verification Mechanism:** A formal protocol for evaluating the reliability of decentralised agents. By verifying high-level predictions against ground-truth physical reality, this mechanism dynamically filters out unreliable peers, ensuring system robustness in open environments without requiring a central authority.

3. **The DT-MADDPG Algorithm:** A hybrid reinforcement learning algorithm that introduces the concept of a "Simulation-Assisted Critic." This contribution theoretically advances the MARL paradigm by demonstrating how a collaborative global simulation model can replace centralised data aggregation, allowing agents to learn near-optimal policies without compromising privacy.

1.6 Thesis Structure

The remainder of this thesis is organised as follows:

- **Chapter 2: Background** establishes the foundational concepts of Digital Twins, Multi-Agent Systems, and Smart Grid.
- **Chapter 3: Literature Review** critically reviews the state-of-the-art, identifying the specific gaps in existing architectures and coordination algorithms.
- **Chapter 4: Framework Design** details the architectural design of the decentralised middleware and the decision-level sharing mechanism.
- **Chapter 5: DT Collab. on Decision Level** introduces the Trust Mechanism and evaluates the benefits of trusted cooperation in a Virtual Power Plant (VPP).
- **Chapter 6: Intelligent Coordination via MARL** presents the DT-MADDPG algorithm, en-

abling agents to move from simple cooperation to optimised, intelligent coordination.

- **Chapter 7: Conclusion and Future Work** summarises the findings, discusses limitations, and outlines future research directions.

Chapter 2

Background

This chapter establishes the foundational knowledge required to understand the core contributions of this thesis. The modern technological landscape is increasingly defined by the challenge of modelling and governing Large Complex Systems, such as smart cities and power grids, whose emergent behaviours presents significant challenges traditional control methods [84]. To address this, the chapter will begin by defining these systems and the fundamental challenges they present. Each section is framed not only to define a concept but also to critically position it in relation to the central research problem of this thesis: resolving the "Privacy-Coordination Paradox" in decentralised systems.

The chapter begins by introducing the concept of Digital Twins, reviewing their origins in Cyber-Physical Systems (CPSs) and examining the architectural paradigms that govern their deployment. This discussion highlights why decentralised "federations" of twins are necessary, yet architecturally incomplete. Following this, the chapter examines the principles of Multi-Agent Reinforcement Learning (MARL), the core methodology for enabling intelligent coordination. This section will critically analyse why standard MARL algorithms are incompatible with the privacy requirements of decentralised systems. Finally, these abstract concepts are grounded in the concrete application domain of Smart Energy Grids, with a specific focus on V2G technology, which serves as the ideal "stress test"

for the proposed framework.

2.1 The Challenge of Large Complex Systems

Large Complex Systems are defined as intricate networks composed of a vast number of interacting components. A defining characteristic of an LCS is emergent behaviour, a phenomenon where the overall macroscopic behaviour of the system cannot be predicted merely by analysing the individual components in isolation. The complex and non-linear interactions between these parts give rise to a system-level dynamic that is significantly more complex than the sum of its constituents [134].

Modern society increasingly relies on the effective functioning of these systems. Prominent examples include smart cities which integrate heterogeneous urban services, national power grids which manage energy generation and distribution across vast geographies, and large-scale urban logistics networks which coordinate thousands of vehicles and deliveries in real-time.

The core properties of LCSs, specifically their immense scale, component heterogeneity, and the high uncertainty resulting from human interactions, present significant challenges for traditional modelling and control strategies [61]. Centralised control methods, which rely on a complete and top-down view of the system, frequently struggle to cope with the immense volume of data and the dynamic, non-stationary nature of these environments. Consequently, there is a pressing need for new paradigms capable of modelling, analysing, and managing these systems effectively. The failure of centralisation is the first pillar of the "Privacy-Coordination Paradox", creating the need for a new architectural paradigm that can manage complexity without a single point of control. This necessity motivates the exploration of Digital Twins, which are discussed in the following section.

Critical Analysis: The failure of traditional methods is not merely a matter of scale, but a fundamental "control paradox." On one hand, simplified models that are computationally tractable fail to capture the rich, non-linear dynamics of the system, leading to suboptimal or even unstable control

decisions. On the other hand, highly detailed, monolithic models that attempt to capture this complexity become computationally intractable for real-time decision-making and create a single point of failure. This paradox reveals that a new architectural approach is needed: one that distributes intelligence rather than centralising it.

2.2 Digital Twins

2.2.1 Cyber-Physical Systems

Cyber-Physical Systems represent the foundational technology that enables the integration of the physical and digital worlds. A CPS is defined as a system in which computational and physical processes are tightly integrated, where embedded computers and networks monitor and control physical processes, usually with feedback loops where physical processes affect computations and vice versa [11, 49].

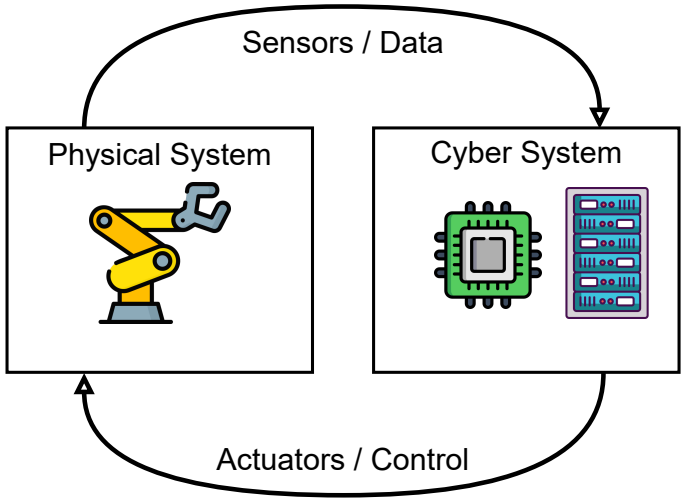

Figure 2.1: A typical Cyber-Physical System architecture.

At the core of a CPS is the interaction between the physical layer and the cyber layer. Sensors in the physical environment collect data regarding the system state, which is transmitted to the cyber layer

for processing. Algorithms within the cyber layer analyse this data and generate control commands, which are subsequently executed by actuators in the physical layer to influence the physical process, as illustrated in Figure 2.1 [1]. This closed-loop mechanism allows for real-time monitoring and automated control of complex infrastructures, such as autonomous automotive systems and industrial process control.

In the context of this thesis, CPS is viewed as the technological predecessor to the Digital Twin. While CPS provides the necessary infrastructure for data acquisition and actuation, traditional CPS frameworks often focus primarily on the immediate control logic and real-time operation. They do not necessarily maintain a high-fidelity, semantic replica of the physical entity for historical analysis or future simulation. To address this limitation and enable proactive "what-if" analysis for complex coordination problems, the Digital Twin emerges as an evolution of the CPS paradigm, specifically by adding a rich virtual model capable of advanced predictive analytics, which enhances this cyber-physical integration [121].

Critical Analysis: The limitation of the traditional CPS paradigm is that it is inherently *reactive*. It senses the current state of the world and acts upon it. This reactive posture is insufficient for optimising complex systems where decisions have long-term consequences. The evolution to the Digital Twin paradigm represents a critical shift from a reactive to a *proactive* or *predictive* stance. By maintaining a high-fidelity model, the Digital Twin can simulate future outcomes, allowing it to make decisions based not just on "what is", but on "what if".

2.2.2 Digital Twins

A Digital Twin is defined as a "live" model or virtual replica of a physical system, which is continuously updated with real-time data from its physical counterpart [37]. This continuous synchronisation enables a bidirectional interaction between the physical and virtual worlds, allowing the DT to serve

[1] Some icons used in this and subsequent diagrams are courtesy of Flaticon.com

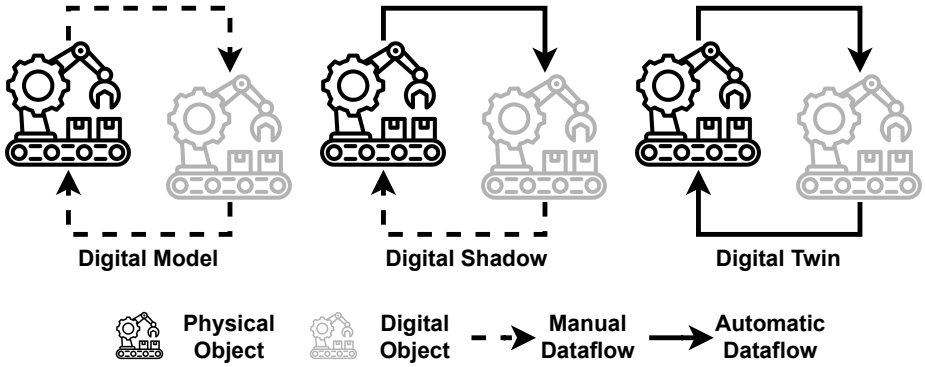

Figure 2.2: The three levels of digital representation: Digital Model, Digital Shadow, and Digital Twin.

not just as a representation, but as a dynamic tool for high-level data analytics and decision support. The primary function of a DT is to enable sophisticated "what-if" analysis, where potential actions can be simulated in the virtual space to assess consequences and identify optimal operational strategies before they are applied to real-world systems [128].

To provide a rigorous definition, it is common in the literature to distinguish between three levels of digital representation: the Digital Model, Digital Shadow, and Digital Twin [36], as illustrated in Figure 2.2.

- **Digital Model:** A digital representation of a physical object with no automated data exchange. The data flow between the physical and digital objects is manual.
- **Digital Shadow:** This advances the concept by introducing a one-way flow of automated data from the physical object to the digital one. The digital model reflects the current state of the real object, but has no direct means to influence it.
- **Digital Twin:** The most advanced form, which completes the cycle with a bidirectional data flow. The DT not only reflects the physical system but can also send control commands back to influence it, forming a closed feedback loop.

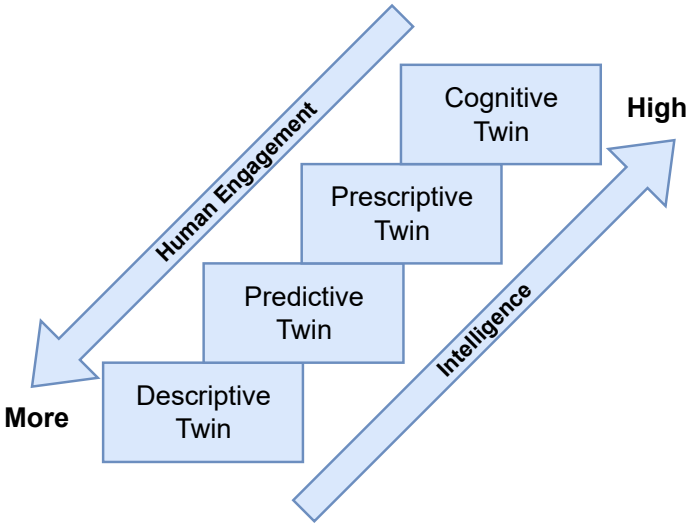

Figure 2.3: Classification of Digital Twins based on intelligence levels.

Digital Twins can also be classified based on their analytics capability and degree of intelligence [149], as shown in Figure 2.3.

- **Descriptive Twin:** Provides information about the physical system as it is now and as it was in the past. It is primarily used for visualisation and monitoring but cannot forecast future states.
- **Predictive Twin:** Utilises predictive analytics, such as machine learning and statistical techniques, to extrapolate what might happen in the future based on historical data.
- **Prescriptive Twin:** Goes beyond prediction by utilising simulation to support "what-if" analysis. It enables the exploration of multiple future scenarios to recommend the best course of action and optimise outcomes.
- **Cognitive Twin:** The most advanced level, capable of replicating human cognitive processes. These twins exhibit self-awareness and can execute conscious, autonomous actions with minimal or no human intervention.

Critical Analysis: While this classification provides a useful roadmap for the capabilities of a single

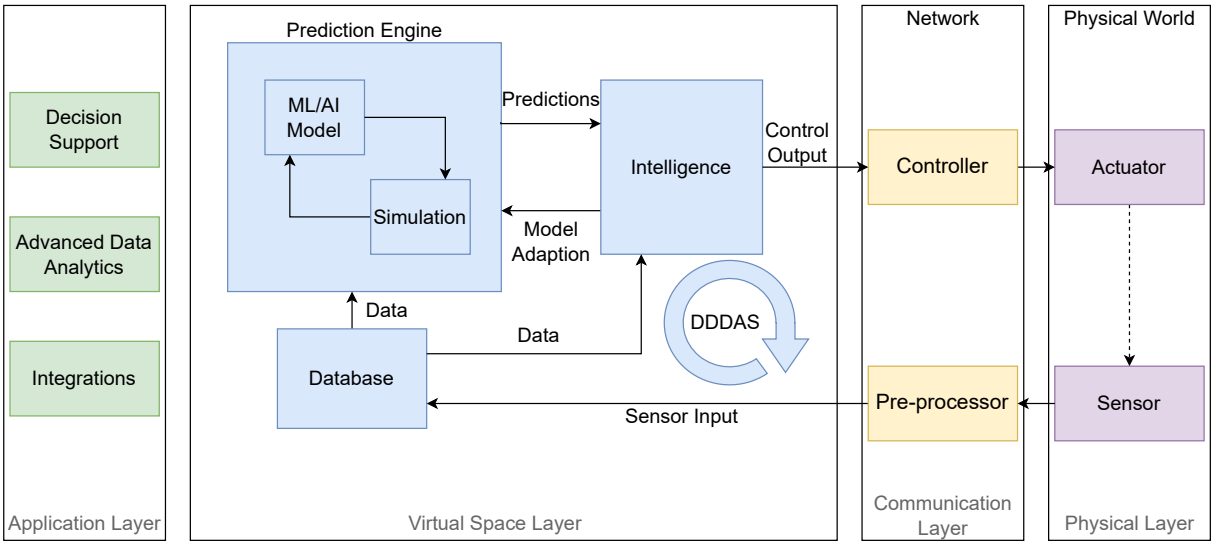

Figure 2.4: A generic layered reference architecture of a Digital Twin.

Digital Twin, much of the existing literature focuses on advancing a single entity up this hierarchy. However, the most pressing challenges in LCSs arise not from the limitations of a single twin, but from the need to coordinate a *network* of them. The true research frontier lies in enabling a federation of less-advanced "Predictive" or "Prescriptive" twins to achieve a collective "Cognitive" intelligence, a problem of coordination rather than individual capability.

A generic layered architecture provides a useful framework for understanding the internal components of a typical Digital Twin, as illustrated in Figure 2.4. This architecture can be conceptualised as four distinct layers:

1. **Physical Layer:** Represents the physical world, containing the requisite hardware components, including *Sensors* for data acquisition and *Actuators* for executing physical changes.

2. **Communication Layer (Network):** Bridges the physical and virtual domains. It utilizes a *Pre-processor* to sanitize and format raw sensor input before transmission, and a *Controller* to reliably relay control outputs to the actuators.

3. **Virtual Space Layer:** The computational core of the system, centered around a **Prediction**

Engine that couples *ML/AI Models* with *Simulation* to forecast system states. A central *Intelligence* module synthesises these predictions with data from the *Database* to execute decision logic.

4. **Application Layer:** Provides high-level interfaces for human operators and external systems, specifically offering modules like *Decision Support*, *Advanced Data Analytics*, and *Integrations*.

The engine of the DT is the feedback loop depicted within the Virtual Space Layer, identified as the Dynamic Data-Driven Application System (DDDAS) cycle [25]. In this process, real-time data flows from the *Pre-processor* into the *Database*, which feeds the *Prediction Engine*. The *Intelligence* module then issues *Control Output* to the network layer and simultaneously triggers *Model Adaption*, ensuring the simulation models are continuously updated to reflect the changing physical environment. This layered structure is relevant to the thesis as it provides the conceptual separation needed for privacy-preserving architectures: raw data can be contained within the lower layers, while only abstracted information from the Virtual Space Layer is shared externally.

2.2.3 Digital Twin Network Architectures

While the generic layered architecture describes the internal components of a single Digital Twin, the physical and logical interconnection of these components across a system defines the **Digital Twin Network (DTN)**. The choice of network architecture has profound implications for the system's scalability, privacy, and robustness [138]. Three primary network paradigms can be distinguished: Centralised, Fully Decentralised, and Hybrid.

The traditional and most prevalent approach is the **Centralised Digital Twin Network**. This paradigm follows a tree-like hierarchy, where data from sensors in the physical world is gathered and transmitted to central servers or a cloud cluster for processing, simulation, and analysis. In this model, the central platform hosts the core of the DTN, maintaining full control over all entities. While this configuration

simplifies management for a single organisation, its reliance on a central authority constitutes a single point of failure and raises significant privacy concerns in large-scale, multi-owner contexts.

In direct contrast, a **Fully Decentralised Digital Twin Network** employs a horizontal architecture analogous to a peer-to-peer system, with no central nodes. All interactions occur directly between peers, and the Virtual Space Layer is distributed across individual nodes (e.g., edge computing nodes) rather than residing on a central server. This model is considered the most privacy-preserving, as no single entity possesses access to or control over all the data in the system. However, achieving system-wide consensus and coordination presents greater challenges in the absence of a central administrator.

The **Hybrid Digital Twin Network** represents a synthesis of the centralised and decentralised paradigms. In this configuration, the system comprises multiple local DTs, where each local DT operates as a self-contained system managing a specific sub-system. These local DTs process data in situ, preserving privacy and enabling local control, while simultaneously retaining the capability to transmit aggregated or high-level information to an upper-level global DT or cloud platform. By blending the local autonomy of a decentralised approach with the potential for global oversight from a centralised one, the hybrid model acts as a "federation" of twins, making it particularly suitable for complex, multi-party systems.

Critical Analysis: The "Federation of Twins" concept is conceptually the most promising for multi-owner LCSs. However, the existing literature often strictly couples this federation to a Hybrid model requiring an upper-level global DT orchestrator. It provides the structure for collaboration but leaves a void where the autonomous decision-making logic should be, frequently forcing a reliance on the central node. This creates the central research gap addressed by this thesis: how to enable these federated twins to learn and coordinate effectively in a *fully decentralised* manner at runtime, without reverting to a centralised control model.

2.3 Principles of Multi-Agent Reinforcement Learning

To enable the "intelligent coordination" proposed in this thesis, we move beyond static rule-based control to adaptive learning systems. This section outlines the principles of Multi-Agent Reinforcement Learning (MARL), which serves as the algorithmic engine for the Digital Twin network.

2.3.1 Reinforcement Learning Fundamentals

Reinforcement Learning (RL) is introduced here as it provides the mathematical framework for learning optimal decision-making policies from interaction, forming the "intelligence" component of the proposed solution. Reinforcement Learning (RL) is a subfield of machine learning concerned with how an intelligent agent ought to take actions in an environment to maximise the notion of cumulative reward. Unlike supervised learning, where the learner is provided with the correct answer, an RL agent must discover which actions yield the most reward by trying them [26].

Formally, an RL problem is often modelled as a Markov Decision Process (MDP), defined by the tuple (S, A, R, P, γ):

- S: The set of possible states representing the environment.
- A: The set of actions available to the agent.
- R: The reward function $R(s,a)$, providing feedback on the action taken.
- P: The transition probability $P(s'|s,a)$, defining the dynamics of the environment.
- γ: The discount factor, determining the importance of future rewards.

The goal of the agent is to learn a *policy* $\pi(a|s)$, a mapping from states to probabilities of selecting each possible action, such that the expected cumulative discounted reward is maximised.

2.3.2 Multi-Agent Systems and MARL

A Multi-Agent System (MAS) is defined as a system composed of multiple interacting, autonomous agents [27]. When these agents use RL to learn their policies simultaneously, the problem becomes Multi-Agent Reinforcement Learning (MARL) [19].

MARL extends the single-agent MDP to a Stochastic Game (or Markov Game). The key distinction is that the state transitions and rewards depend on the *joint* action of all agents, not just an individual one. This makes it the natural paradigm for solving the *coordination* half of the "Privacy-Coordination Paradox," but also introduces significant challenges:

- **Non-Stationarity:** From the perspective of any single agent, the environment is unstable because other agents are constantly updating their policies. This violates the stationarity assumption required by most single-agent RL algorithms (like Q-Learning), often leading to failure in convergence [28].
- **Scalability:** The joint action space grows exponentially with the number of agents, making standard approaches computationally intractable for Large Complex Systems.

2.3.3 The CTDE Paradigm

To address the challenges of non-stationarity and coordination, a dominant framework in modern MARL is Centralised Training with Decentralised Execution (CTDE) [5]. This paradigm is most commonly implemented using an *actor-critic* architecture [65], which is essential for understanding its power and its limitations.

In this structure, each agent is composed of two parts:

- **The Actor:** This is the agent's policy, $\pi(a_i|o_i)$, which maps its local observation o_i to an action a_i. During execution, the actor is all that is needed.

- **The Critic:** This is a value function, typically an action-value function $Q_i(s, a_1, \ldots, a_N)$, that estimates the expected future reward given a state and the *joint action* of all agents. The critic's role is to provide a stable and informative learning signal to the actor.

The CTDE paradigm leverages this structure by separating the information available during training and execution:

1. **Centralised Training:** During the training phase, which can be performed offline in a simulator, the critics are allowed access to global information. The critic for agent i, Q_i, can observe the full state of the environment and the actions of all other agents. This global view is critical because it allows the critic to properly solve the *credit assignment problem*: determining how much each agent's individual action contributed to the collective outcome. It also makes the learning environment stationary from the critic's perspective, which is essential for stable convergence.

2. **Decentralised Execution:** During the execution (deployment) phase, the centralised critics are discarded. The trained actors are deployed to the real-world agents (e.g., the Digital Twins). These actors operate using only their local observations, making the system fully decentralised at runtime.

This paradigm is particularly well-suited for the Digital Twin networks proposed in this thesis: training can occur in a rich, data-heavy simulation environment (the "Centralised" phase), while the resulting policies are deployed to independent, privacy-preserving nodes in the real world (the "Decentralised" phase). Common algorithms utilizing this paradigm include Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [79], which will be discussed in Chapter 6.

Critical Analysis: Despite its power, a critical architectural mismatch exists when applying the standard CTDE paradigm to multi-owner Digital Twin networks. The core assumption of CTDE that the centralised critic can access the complete, raw state and action information of all agents during

training directly conflicts with the privacy and data sovereignty requirements of the domain. An EV owner, for example, would be unwilling to share their precise location, battery data, and intended actions even for an "offline" training process controlled by a third party. This creates a methodological impasse: system designers are forced to either abandon the high-performance coordination of CTDE in favour of less effective, fully decentralised learning methods, or violate the privacy principles that are essential for user participation. This conflict necessitates a novel adaptation of the CTDE framework, one that can provide the critic with the necessary global context without violating privacy. Resolving this impasse is a core contribution of this thesis.

2.4 Application Domain: Smart Energy Grids

This thesis validates the proposed Digital Twin framework within the domain of Smart Energy Grids. This domain is selected because it represents a prime example of a Large Complex System, characterised by distributed assets, high volatility, and a critical need for privacy-preserving coordination.

2.4.1 Overview of Smart Energy Grids

A Smart Grid is an electricity network that utilises digital Information and Communication Technology (ICT) to monitor and manage the transport of electricity from all generation sources to meet the varying electricity demands of end-users [80, 56]. Unlike traditional power grids, which generally facilitate a one-way flow of electricity and information from central power plants to consumers, Smart Grids enable a **bidirectional flow** of both energy and data [137].

This bidirectional capability allows for the integration of renewable energy sources, automated maintenance, and real-time demand response. However, it also introduces significant complexity, as the grid must now coordinate millions of active endpoints rather than a few dozen central generators. This complexity, combined with the involvement of private asset owners, makes the Smart Grid the ideal

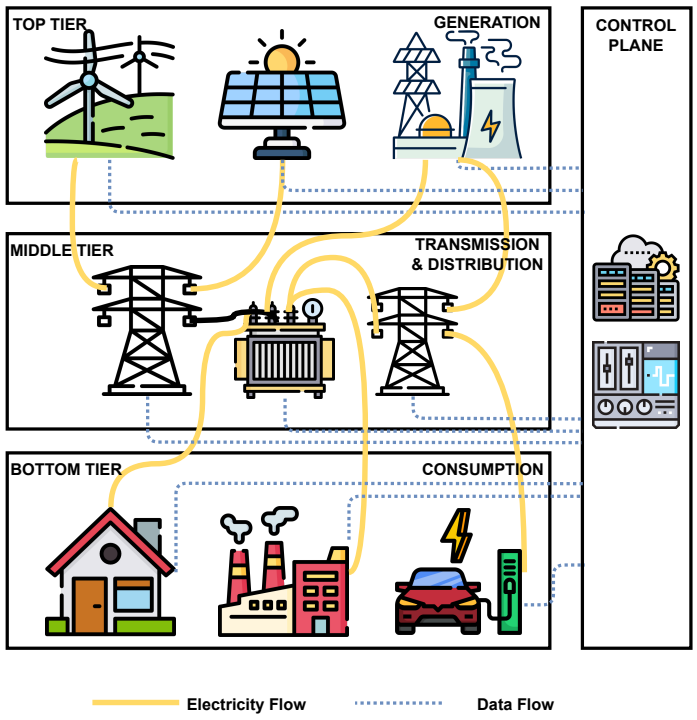

Figure 2.5: A typical Smart Grid architecture showing bidirectional energy and data flows.

application domain to test solutions for the Privacy-Coordination Paradox.

2.4.2 Virtual Power Plants

A key operational model within the Smart Grid is the Virtual Power Plant (VPP). A VPP is a cloud-based distributed power plant that aggregates the capacities of heterogeneous Distributed Energy Resources (DERs), such as rooftop solar panels, wind turbines, and battery storage systems, for the purposes of enhancing power generation, trading energy, and ensuring grid reliability [102]. By coordinating these geographically dispersed assets, a VPP can function as a single, dispatchable power plant. This allows small-scale producers to participate in energy markets that were previously accessible only to large utility companies.

Critical Analysis: The VPP model is a direct manifestation of the "Privacy-Coordination Paradox".

The VPP operator requires reliable, predictable power (coordination), which implies a need for visibility into the state of each asset. However, the asset owners (e.g., homeowners with solar panels) require autonomy and privacy over their resources. This tension between the operator's need for global optimisation and the owner's right to local control makes the VPP a perfect microcosm for the central problem this thesis aims to solve.

2.4.3 Vehicle-to-Grid Technology

V2G technology represents a specific and highly dynamic subset of the VPP concept. V2G describes a system in which plug-in EVs communicate with the power grid to sell demand response services by either returning electricity to the grid or by throttling their charging rate [119].

In a V2G network, EVs are transformed from passive energy consumers into active, mobile energy storage units, as illustrated in Figure 2.6. They can be scheduled to charge during low-demand "valley" times (e.g., overnight) and discharge stored energy back to the grid during high-demand "peak" times. This capability provides essential grid services such as peak shaving, load leveling, and frequency regulation. However, V2G presents a unique coordination challenge: it involves managing thousands of autonomous agents (EV owners) who have unpredictable mobility patterns and strict privacy requirements regarding their location and usage habits. This thesis utilises V2G as the primary use case to demonstrate the effectiveness of privacy-preserving, intelligent Digital Twin coordination.

Critical Analysis: If the VPP is a microcosm of the coordination problem, then V2G is its most extreme and challenging manifestation. The assets are not only decentralised but also mobile, and the owners are not commercial entities but private citizens, making privacy concerns paramount. The high stochasticity of user behaviour and mobility pushes traditional control models to their breaking point. Therefore, V2G is not just a convenient application domain; it is the ideal "stress test" for any proposed decentralised coordination framework. A solution that is robust, private, and scalable enough for V2G is likely to be generalisable to a wide range of other LCSs.

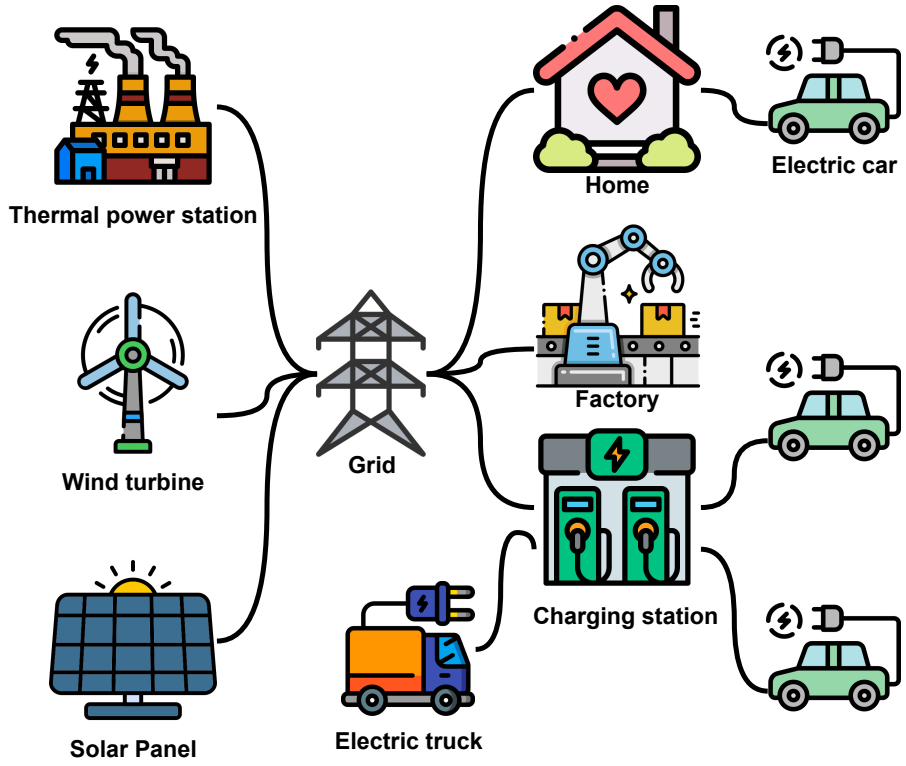

Figure 2.6: An example of a V2G system, where electric vehicles interact with the grid for bidirectional energy flow.

2.4.4 Technical Requirements and Domain Suitability

The selection of Smart Energy Grids, and the V2G paradigm in particular, as the primary application domain is not merely due to its industrial relevance. It is chosen because it embodies a hierarchy of challenges that serve as a rigorous "stress test" for any decentralised coordination framework. These challenges span from fundamental data governance constraints to socio-technical incentive problems and complex environmental dynamics, making it the ideal testbed for the framework proposed in this thesis.

The Criticality of Data Sovereignty

Smart energy data is uniquely sensitive because it acts as a high-resolution proxy for human activity. Inferences regarding a household's occupancy, a driver's daily commute, and even socioeconomic status can be derived from granular energy consumption and discharge profiles. In a multi-stakeholder environment involving private citizens, commercial fleet operators, and utility providers, the mandate for *Data Sovereignty* is not just a preference but a legal and social requirement (e.g., GDPR compliance) [46]. This domain, therefore, provides a realistic environment to validate the "Privacy-Coordination Paradox," as any successful solution must achieve system-wide stability without the luxury of raw data aggregation.

The Socio-Technical "Incentive Gap"

Smart Grids represent a complex socio-technical system where global objectives (e.g., frequency regulation and peak shaving) are often at odds with the local utilities of individual agents (e.g., maximizing battery longevity or ensuring a full charge for a morning commute) [99]. This creates an "Incentive Gap." A centralised controller might optimise for the grid at the expense of the individual, leading to a lack of participant "willingness to cooperate." By utilizing this domain, this research can demonstrate how decentralised Digital Twins can bridge this gap through a "Trust-Based" mecha-

nism, aligning individual self-interest with the collective good through intelligent, privacy-preserving negotiation.

Stochasticity and High-Frequency Dynamics

Finally, beyond the socio-technical constraints, the environment itself is highly dynamic. Unlike traditional industrial systems that operate under relatively stable set-points, the modern Smart Grid is subject to extreme stochasticity. This volatility is driven by the intermittency of renewable energy sources and the unpredictable behavioral patterns of human participants. In a V2G context, an agent's availability is governed by mobility needs that are often decoupled from the grid's state. Consequently, any coordination framework must be capable of managing a *dynamic topology* where nodes (EVs) join and leave the network asynchronously. This necessitates a move away from static, centralised optimisation models toward the adaptive, real-time learning capabilities of MARL.

2.5 Summary

This chapter has established the foundational knowledge required for this thesis. It defined the problem of **Large Complex Systems** and the need for decentralised modeling. It presented **Digital Twins** as the core enabling technology, distinguishing between centralised and decentralised network architectures. It introduced **Multi-Agent Reinforcement Learning** and the **Centralised Training with Decentralised Execution** paradigm as the method for enabling intelligent coordination. Finally, it described the Smart Grid and V2G domains as the testing ground for these concepts.

With these foundations established, the next chapter will critically review the existing literature to identify why current state-of-the-art solutions fall short in enabling secure, intelligent coordination within these decentralised environments.

Chapter 3

Literature Review

3.1 Introduction

This chapter establishes the academic context for the thesis by critically reviewing literature across three intersecting domains: Digital Twin architectures, collaborative frameworks for distributed systems, and intelligent coordination methodologies. While the concept of the Digital Twin has matured significantly, its application to Large Complex Systems such as Smart Cities and national Energy Grids remains hindered by a fundamental conflict between **coordination** and **privacy**. The review aims to identify the specific gaps that necessitate the development of the decentralised, trust-based, and intelligent framework proposed in this thesis.

To provide a comprehensive analysis, the chapter is structured as follows:

- **Section 3.2** critically evaluates the dominant centralised Digital Twin paradigm, highlighting its inherent limitations in scalability, latency, and privacy when applied to multi-owner Large Complex Systems.
- **Section 3.3** reviews emerging decentralised alternatives, including edge-based architectures

[120], Federated Learning [113], and peer-to-peer frameworks [110]. It critically analyses the specific limitations of each approach, from the coordination gap in architectural solutions to the methodological mismatch of statistical learning paradigms.

- **Section 3.4** explores the challenge of "Soft Trust" and reputation in decentralised networks [68], distinguishing behavioural reliability from cryptographic security and critiquing the limitations of existing reputation models for Digital Twin data.
- **Section 3.5** reviews state-of-the-art methodologies for coordinating V2G systems [118], focusing on both conventional optimisation techniques and Multi-Agent Reinforcement Learning [19]. It critiques the privacy assumptions underlying these methods.
- **Section 3.6** synthesises the findings from the preceding sections to precisely define the research gap addressed by this thesis.

Ultimately, this analysis identifies the specific research gap: the lack of a unified framework that enables privacy-preserving, trusted, and intelligent coordination among decentralised agents.

3.2 Limitations of Centralised Digital Twin Architectures

The deployment architecture of a Digital Twin significantly influences its capability to manage Large Complex Systems.

The Dominant Paradigm: The traditional and most prevalent deployment model for Digital Twins follows a centralised, tree-like hierarchy [110]. This paradigm, rooted in early NASA missions and the initial wave of Industry 4.0, conceptualises the Digital Twin as a "Single Source of Truth" residing in a high-performance computing environment. In this architecture, data from physical assets is gathered and transmitted to a central platform,btypically a cloud cluster such as AWS IoT TwinMaker [6] or Microsoft Azure Digital Twins [86] that hosts the core components, including high-fidelity models,

historical databases, and simulation engines [4].

This paradigm simplifies management for single-owner systems by providing a unified point of control and aggregation, facilitating global state monitoring and complex cross-asset analytics. By consolidating all telemetry into a monolithic data lake, the central authority can perform exhaustive optimisations that are theoretically optimal for the entire system. However, while this approach is effective for closed environments like a single manufacturing plant under a single administrative domain, it introduces two prohibitive limitations for Large Complex Systems involving multiple stakeholders and geographically dispersed assets.

Scalability and Latency Constraints: For Large Complex Systems such as smart cities or national power grids, this centralised approach faces severe scalability bottlenecks. As noted by Cakir et al. [20], transmitting high-fidelity real-time data such as video streams or millisecond-level sensor readings from thousands of edge assets (e.g., EVs, smart meters) to a central server creates prohibitive bandwidth requirements that scale linearly or even exponentially. In scenarios like smart urban logistics, aggregating all raw data from a city-scale fleet for real-time analytics often becomes computationally and communicationally non-viable [60]. Furthermore, the round-trip latency inherent in this architecture such as sending data to the cloud, processing it, and returning a control signal, often exceeds the strict millisecond-level constraints required for critical tasks like grid frequency regulation or autonomous vehicle control [93]. Consequently, centralised architectures effectively act as a bottleneck for real-time responsiveness and introduce a critical single point of failure for infrastructure management.

Privacy and Security Vulnerabilities: The most significant limitation is the assumption of a single data owner. In multi-stakeholder environments like urban logistics or V2G networks, data is dispersed among distinct entities such as private vehicle owners, fleet operators, and grid utilities who are often legally or commercially restricted from sharing raw data [67, 135]. Centralised architectures, which demand the surrender of sensitive operational data (e.g., driving patterns, location history) to a central

authority, fundamentally conflict with these constraints [17]. This requirement not only creates a "privacy barrier" that disincentivises participation and risks violating regulations such as GDPR [38], but also establishes a single point of failure and a high-value target for cyberattacks, making such implementations impractical for broad public services.

3.3 State of the Art in Collaborative Digital Twin Frameworks

To address the limitations of centralised architectures, recent research has explored various methodologies for enabling collaboration among distributed Digital Twins. These approaches generally fall into three categories: decentralised architectural patterns, federated learning integration, and peer-to-peer interaction frameworks. Each subsection will conclude with a critical analysis of the limitations of the reviewed approaches, systematically building the case for the contributions of this thesis.

3.3.1 Decentralised and Edge-Based Architectures

In response to scalability challenges, research has diverged into two main streams of decentralisation: the distribution of data and the distribution of simulation or instruction logic. Regarding **data distribution**, methods focus on managing dispersed resources to create a cohesive information layer. For instance, Moshrefzadeh et al. [90] introduced the concept of a "Distributed Digital Twin" in the agricultural landscape. Their approach handles resources from different stakeholders by integrating information from physical assets using a metadata model. While effective for data availability, this relies on a service-oriented architecture that focuses more on information access than on autonomous local simulation.

Conversely, regarding **simulation and instruction distribution**, recent paradigms emphasise distributing the computational logic and decision-making capabilities. Vergara et al. [130] and Vergara et al. [131] proposed the "Federated Digital Twin" framework, which enables the interconnection of

fully autonomous DTs in virtual space. This moves beyond simple data sharing to facilitate valid global decision-making through purely peer-to-peer interactions. Hierarchical approaches also fall into this category; Villalonga et al. [133] designed a framework where local DTs execute local control decisions while aggregating insights to a global DT for high-level scheduling.

Building on this distributed logic, Infante et al. [48] introduced the OpenTwins platform, redefining the architecture to be lightweight and synchronised across diverse infrastructures, including ARM-based Internet of Things (IoT) devices. Similarly, Costantini et al. [24] proposed the IoTwins platform, which employs a hybrid model integrating edge and cloud computing to feed AI services, demonstrating how distributed platforms can leverage heterogeneous resources.

To address the challenge of integrating heterogeneous and physically separated twins, several frameworks propose an ecosystem approach. Yu et al. [145] introduced the Internet of Federated Digital Twins (IoFDT), a hierarchical architecture bridging cyber and physical spaces through horizontal and vertical interactions. This concept is echoed by Liu et al. [77] with the Internet of DT (IoDT), which focuses on collaborative resource sharing and integrates advanced communication technologies like 5G/6G. Moving towards web standards, Ricci et al. [107] proposed the "Web of Digital Twins" (WoDT), an open ecosystem for connected twins independent of specific domains. Giulianelli et al. [34] extended this with the HWoDT framework, providing a toolchain based on web standards to develop ecosystems of heterogeneous assets. Some architectural approaches explicitly leverage learning paradigms to facilitate this distribution; for example, San et al. [111] utilised federated learning as the basis for a Decentralised Digital Twin (DDT) framework, enabling cooperative model learning without exposing client-specific data. While these models improve scalability by offloading computation, hierarchical versions often retain a dependency on a global coordinator, which may be a bottleneck in fully decentralised environments.

A distinct subset of this decentralised paradigm involves Edge Digital Twins, where the digital replica resides directly on the edge computing infrastructure or the physical device itself. Picone et al. [100]

presented the Edge Digital Twins (EDT) model, enabling lightweight replication on devices to support autonomous collaboration. By co-locating data processing with the asset, these architectures reduce control loop latency. This is particularly critical in time-sensitive domains; Kondo et al. [66] proposed a Local Digital Twin (LDT) architecture for manufacturing that meets strict timing constraints by processing at the edge, while maintaining a link to a Global DT for oversight. Similarly, in the automotive domain, El Azzaoui et al. [30] developed a digital twin framework on the intelligent edge for Internet of Vehicle (IoV) systems, optimizing communication and reducing the carbon footprint by offloading computation.

From a data perspective, this edge-centric approach inherently preserves privacy, as raw sensor streams such as high-frequency voltage readings or driver behaviour logs are processed locally, with only aggregated insights or specific events transmitted to the wider network [120]. However, this introduces a trade-off between the fidelity of the simulation and the available computational resources at the edge, necessitating lightweight modelling techniques or hybrid cloud-edge orchestration.

Critical Analysis: While decentralised and edge architectures successfully address the scalability and privacy failures of centralised models, they primarily solve the *architectural* problem of infrastructure and the *interoperability* problem of data exchange. Frameworks like the Web of Digital Twins or IoFDT provide the necessary "pipes" and standard languages for agents to communicate, but they do not inherently solve the *coordination* problem. By distributing intelligence to the edge, these architectures risk creating "information silos" where agents have a perfect view of their local state but lack the global context required for system-wide optimisation. Furthermore, many proposed hierarchical solutions still rely on a global coordinator for high-level decision-making, reintroducing potential bottlenecks. There remains a significant gap in the "intelligence" required for fully decentralised agents to learn how to align their actions effectively without reverting to centralised control.

3.3.2 Federated Learning Approaches

Federated Learning (FL) has emerged as a pivotal technology for facilitating collaboration among Digital Twins while rigorously enforcing data privacy. In this paradigm, individual DTs train machine learning models locally on their specific operational data such as user behaviour or private sensor logs and share only the model updates (gradients or weights) with a central server or peer network. This mechanism ensures that sensitive raw data never leaves the local environment, adhering to strict privacy regulations [83, 97]. Furthermore, FL empowers individual DTs with collective intelligence; by aggregating insights from a vast network of peers, a local DT can enhance its own decision-making capabilities, improving predictive maintenance or anomaly detection by leveraging a global knowledge base that would be impossible to construct in isolation [111].

The versatility of FL in enhancing Digital Twins has been demonstrated across diverse domains. In the context of smart cities, Ramu et al. [104] surveyed the integration of FL with DTs, highlighting its capacity to ensure trustworthiness and privacy in life-critical urban applications such as transportation and environmental management. This utility extends to industrial settings; Padmavathi et al. [96] proposed a method combining Edge AI and FL for smart factories, achieving significant reductions in latency (35%) and cloud usage (28%) by enabling edge devices to learn collaboratively without exposing proprietary production data.

In domains where data sensitivity is paramount, such as healthcare, FL serves as a critical enabler. Nagaraj et al. [92] identified FL as a unique solution for clinical Digital Twins, allowing for the scaling of precise medical models without compromising patient confidentiality. Similarly, Hemdan and Sayed [41] demonstrated the practical application of this synergy for neurological disorder detection, using FL to classify EEG signals within a secure, blockchain-enhanced framework.

The paradigm is also particularly relevant for dynamic, mobile networks. Khan et al. [63] outlined an architecture for FL-enabled Digital Twins in vehicular networks, facilitating resource management

and infotainment services while preserving user privacy. Addressing the complexity of coordination in such networks, Sun et al. [114] explored air-ground networks where drone-based DTs act as aggregators, employing game-theoretic incentive schemes to reconcile the conflict between privacy protection and the need for collaborative data training.

However, despite these advantages, a critical limitation of FL in the context of Digital Twins is the opaque nature of the resulting models. FL typically produces "black-box" deep neural networks which, while effective for pattern recognition and prediction, lack the inherent interpretability of physics-based or logic-based simulations [150]. This opacity poses a significant challenge for industrial decision support: while an FL-based DT might predict that a failure will occur, it often cannot explain why, nor can it reliably support the sophisticated "what-if" simulations required to test potential interventions. Consequently, relying solely on FL can decouple the DT from its mechanistic roots, reducing it to a statistical estimator rather than a transparent simulation tool.

Critical Analysis: Beyond the interpretability challenges, FL presents distinct architectural and security hurdles for decentralised DTs. Although it avoids raw data transfer, the transmission of high-dimensional model gradients incurs substantial *communication overhead*, which can be prohibitive for resource-constrained edge devices operating over bandwidth-limited networks [57]. Moreover, the privacy guarantee is not absolute; shared gradients remain vulnerable to "model inversion" attacks, potentially allowing adversaries to reconstruct sensitive training samples [154]. Most critically, FL is fundamentally a technique for *distributed training*, not *distributed coordination*. It enables agents to learn a shared predictive model, but it does not provide the mechanisms for autonomous agents to negotiate, align their physical actions, or resolve conflicts in real-time. It solves the learning problem, but leaves the actuation and coordination gap unaddressed.

3.3.3 Peer-to-Peer Interaction Frameworks

Peer-to-Peer (P2P) technologies have been extensively explored to establish trust and facilitate direct interaction in decentralised collaboration. A significant portion of this research focuses on blockchain technology. Sahal et al. [109] proposed a blockchain-based framework for decentralised epidemic alerting, enabling DTs to share historical data securely. Similarly, García-Valls and Chirivella-Ciruelos [33] introduced CoTwin, utilizing blockchain to create an immutable storage space for collaborative model training. In the industrial domain, Tao et al. [122] and Li et al. [71] proposed blockchain-enhanced mechanisms to manage distributed manufacturing resources and address interaction gaps. Putz et al. [103] extended this with the entire value chain with EtherTwin, a decentralised application for secure information management.

Beyond basic data sharing, blockchain has been applied to specific governance challenges. Redeker et al. [106] presented a platform for Industry 4.0 that combines blockchain with distributed file systems to collaboratively enrich Asset Administration Shells while ensuring data sovereignty. Addressing intellectual property, Li et al. [72] proposed a schema for preserving DT copyrights in social manufacturing networks. Furthermore, a recent survey by Li et al. [69] highlighted the potential of P2P payment channels, such as the lightning network, to support micro-transactions for real-time data queries between twins.

While blockchain focuses on the ledger, other frameworks emphasise the communication layer itself. Li et al. [70] explored the concept of a DTN where virtual agents spontaneously discover and exchange information using P2P protocols, independent of the underlying physical infrastructure. To handle information dissemination in these massive-scale, volatile systems, gossip protocols offer a lightweight alternative to heavy consensus mechanisms. Kempe et al. [62] established the foundation for using gossip protocols to compute aggregate functions like sums and averages with exponential convergence speed. Recent applications have adapted this for advanced tasks; Husnoo et al. [47] utilised Random Walk gossip protocols to decentralise FL for anomaly detection in power systems,

effectively removing the central aggregator. Similarly, Azimi and Sajedi [10] employed the gossip-based VICINITY protocol to facilitate distributed data clustering, allowing nodes to build summary models of the global dataset through local exchanges.

Critical Analysis: The landscape of P2P interaction frameworks reveals a dichotomy between security and agility. Blockchain-based solutions offer robust *transactional integrity* and auditability, effectively solving the "trust" problem for historical data. However, they are ill-suited for *behavioural coordination* in real-time systems; the latency and throughput bottlenecks inherent in consensus mechanisms render them impractical for high-frequency control loops. Conversely, lightweight protocols like gossip enable scalable information dissemination but typically operate at the level of raw data aggregation or simple state synchronisation. They lack the semantic structure to support the negotiation of complex future *intentions*. Ultimately, while these technologies provide the necessary connectivity "pipes" and "ledgers," they do not inherently enable agents to align their future actions intelligently without exposing sensitive internal states. They provide a "system of record" or a "system of transport," but not a "system of coordination."

3.4 Trust and Reputation in Decentralised Networks

As Digital Twin networks transition from closed, enterprise-controlled environments to open, multi-stakeholder ecosystems, the assumption of good intentions among participating agents becomes invalid. In a decentralised V2G network, agents are owned by self-interested individuals who may have incentives to manipulate information, for example, reporting a lower State of Charge (SoC) to receive prioritised charging. Consequently, establishing a "soft trust" layer is as critical as the "hard security" provided by encryption.

3.4.1 The Distinction Between Security and Trust

It is imperative to distinguish between "Hard Security" and "Soft Trust." Existing literature in Industrial IoT largely focuses on the former, employing cryptographic mechanisms such as Public Key Infrastructure (PKI) and Elliptic Curve Cryptography (ECC) to ensure authentication and data integrity. However, these mechanisms only verify the identity of the sender and the technical integrity of the transport; they do not verify the *veracity* of the information content itself.

The theoretical underpinnings of this distinction are well-established. Momani and Challa [89] emphasised that while often conflated in network domains, security and trust are distinct; security focuses on protection against threats often via heavy overheads, whereas trust deals with the reliability of interactions. Jensen [50] further articulated this by classifying trust as a "soft" security property, warning that "hard" security mechanisms which rely on implicit trust assumptions are prone to failure if that trust is misplaced.

Historically, the concept of trust in computer science emerged from security research. Artz and Gil [9] observed that early approaches treated trust as a derivative of "hard security," established through authentication and access control policies. Blaze et al. [16] advanced this by proposing "Trust Management" engines that move away from simple identity resolution towards a unified policy approach, where authorisation is granted based on verified credentials and privileges rather than just identity.

In this thesis, **Soft Trust** is defined as the quantifiable measure of an agent's behavioral reliability [50]. A digitally signed message stating "I have 50kWh available" is cryptographically secure (Hard Security) but functionally unreliable (Soft Trust) if the physical battery is actually empty due to a sensor fault or model drift. This research addresses the challenge of evaluating the trustworthiness of the *content* in the face of such uncertainty, moving beyond identity verification to behavioral verification.

3.4.2 Reputation Management in Decentralised Systems

To manage this uncertainty, a common approach in decentralised systems is the use of reputation systems, which aggregate historical interactions to predict future behaviour [54]. These systems range from centralised models, such as eBay's feedback score, to fully decentralised algorithms like EigenTrust, which was designed for P2P file-sharing networks [59]. Many of these systems are probabilistic, using models like the Beta Reputation System to represent trust as a probability distribution based on past positive and negative interactions [53].

Building on these foundations, Xiong and Liu [139] introduced PeerTrust, a framework that quantifies trustworthiness based on transaction feedback and community context. However, these systems are susceptible to manipulation. Hoffman et al. [42] and Swamynathan et al. [117] extensively surveyed vulnerabilities such as collusion and the use of short-lived identities to whitewash bad reputations. Furthermore, Hasan [40] highlighted that privacy concerns often deter users from providing truthful negative feedback, undermining the system's accuracy.

In the context of IoT and edge computing, recent research has integrated advanced technologies to address these limitations. Ullah et al. [126] proposed the DeepTrust framework, leveraging deep learning to identify trustworthy devices with significantly higher precision than conventional models. Blockchain technology has also been widely adopted to ensure the integrity of reputation data; Battah et al. [14] utilised smart contracts for a reward-penalty scheme, while Meimin Wang [85] developed a multi-level reputation evaluation for the Internet of Everything. Addressing the resource constraints of industrial edge devices, Ud Din et al. [124] introduced LightTrust, which employs a centralised trust agent to manage certificates, reducing the computational burden on individual nodes.

A persistent challenge in these systems is handling dynamic behaviour, where an agent's reliability changes over time. Gómez Mármol and Martínez Pérez [39] demonstrated that static models like EigenTrust degrade in performance under dynamic network topologies. Addressing this, Player and

Griffiths [101] proposed the RaPTaR method, which outperforms static sliding-window techniques in tracking agents with fluctuating behaviour patterns.

Critical Analysis: While existing reputation systems effectively manage trust in service-oriented or social networks, they exhibit fundamental limitations when applied to DTs. First, there is a *data mismatch*: most models are designed for discrete, binary feedback (e.g., success/failure), rendering them incompatible with the continuous, high-dimensional data streams inherent to physical simulations. Second, they typically rely on *subjective consensus*, aggregating peer opinions which are vulnerable to collusion. This ignores the unique advantage of DTs: the availability of objective physical ground truth. Finally, these models often fail to distinguish between malicious intent and non-malicious incompetence (e.g., sensor drift), a critical distinction for maintaining system stability without unfairly penalising faulty but honest nodes. Consequently, a bespoke trust mechanism is required that anchors reputation in physical reality rather than social consensus.

3.5 A Critical Review of V2G Coordination Methodologies

The challenge of coordinating large-scale V2G systems has been approached through various methodologies, ranging from classical mathematical optimisation to advanced data-driven learning.

3.5.1 Conventional Optimisation Methods

A significant body of work has applied game theory and mathematical optimisation to V2G coordination. Game theoretic approaches have been widely explored to model the dynamic interactions between stakeholders. Marousi and Charitopoulos [81] surveyed the transition from centralised to decentralised decision-making in energy systems, highlighting the utility of game theory in capturing multi-stakeholder dynamics. Specific implementations include Zhang et al. [151], who formulated the problem as a mean field game to propose a distributed control algorithm based on equilibrium strate-

gies, and Zhou et al. [153], who presented an aggregative game framework to model the decision-making of heterogeneous EVs.

Beyond game theory, classical optimisation remains a common tool. Yoon et al. [143] formulated charging coordination as an integer linear programming problem. To address the intermittency of renewable energy, Nguyen et al. [94] proposed a control strategy using optimisation to coordinate both charging and discharging. However, these methods often struggle with uncertainty. Chai et al. [22] introduced a two-stage optimisation method combining day-ahead planning with real-time adjustment, while Anastasiadis et al. [7] developed a stochastic multiobjective economic dispatch model that incorporates penalties for the stochastic behaviour of wind power and V2G stations. Lotfi et al. [78] further proposed a robust optimal coordinated charging model based on linear programming to handle market uncertainties.

Critical Analysis: Conventional optimisation methods suffer from three primary limitations. First, they are **brittle**; their performance is highly dependent on the accuracy of a pre-defined system model, which is often a simplified representation of reality and cannot capture the full non-linear dynamics of an LCS. Second, they are **static**; they solve for an optimal policy based on a fixed model and cannot adapt online to changes in the environment or user behaviour without manual re-engineering. Finally, many of these methods, particularly those involving integer programming or exhaustive search, are **computationally intractable** at the scale of thousands or millions of agents, limiting their real-world applicability.

3.5.2 Multi-Agent Reinforcement Learning

To overcome the limitations of static models, Multi-Agent Reinforcement Learning (MARL) has gained prominence. MARL enables agents to learn complex policies through trial-and-error interaction, making it highly suitable for dynamic environments. Foundational algorithms such as QMIX [105] and MAPPO [144] have demonstrated that decentralised policies can be trained effectively in

complex multi-agent settings, providing a strong baseline for cooperative tasks.

In the context of Smart Grids, MARL has been extensively applied to various control problems, as reviewed by Li et al. [73] and Al-Saadi et al. [2]. For grid stability, Yan and Xu [140] utilised Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for load frequency control, effectively minimising control errors against stochastic frequency variations. Similarly, Hu et al. [43] proposed an experience-augmented multi-agent actor-critic algorithm for voltage control, coordinating solar and storage systems to mitigate fluctuations. Addressing resilience, Kamruzzaman et al. [58] developed a hybrid soft actor-critic algorithm to plan the deployment of shunts, enhancing the system's ability to withstand line outages.

Specific to V2G and EV coordination, Park and Moon [98] applied MADDPG and COMA algorithms to schedule charging for multiple vehicles, demonstrating the suitability of decentralised execution for real-time requirements. Beyond physical control, MARL is also used for market-based coordination. Roesch et al. [108] demonstrated that MARL-based control of industrial sites outperforms rule-based strategies in optimising energy demand and supply. Aladdin et al. [3] developed MARLA-SG for efficient demand response, achieving significant cost reductions. Furthermore, Fang et al. [31] applied a Multi-Agent Deep Q-Network (MADQN) to generate real-time market decisions in microgrids, while Barth et al. [13] investigated distributed learning algorithms to achieve Nash equilibria for energy independence.

A dominant paradigm in this field is Centralised Training with Decentralised Execution (CTDE). Methods like MADDPG[79] utilise a centralised critic to observe joint actions and provide stable learning signals to decentralised actors. Other approaches focus on value decomposition, such as Value-Decomposition Network (VDN)[115], which factorises a team's joint value function into agent-wise components to solve the credit assignment problem. Additionally, techniques like Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL) [32] allow agents to learn explicit coordination by passing gradients through a differentiable communication channel.

Critical Analysis: While MARL offers a robust framework for dynamic coordination, its application to critical infrastructure is constrained by three fundamental challenges. First, deep reinforcement learning is notoriously **sample inefficient**, often requiring millions of interactions to converge, which necessitates high-fidelity simulation environments that may not always be available. Second, the **non-stationarity** inherent in multi-agent learning where the environment changes as other agents learn can lead to unstable convergence and unpredictable emergent behaviours. Finally, and most significantly for this research, the reliance on **centralised training** mechanisms (as seen in CTDE) creates a privacy bottleneck. The requirement to aggregate global state information during training conflicts with the data sovereignty of autonomous agents, a specific architectural tension that is examined in the following section.

3.5.3 The Privacy-Coordination Gap in MARL

Despite the power of MARL, a fundamental conflict exists when applying it to privacy-sensitive DT networks. State-of-the-art algorithms like MADDPG typically assume that the centralised critic has direct access to a centralised database containing the raw state and action information of all agents [79].

To illustrate this with a concrete example, consider a grid operator using MADDPG to train a V2G coordination policy. To update its centralised critic, the algorithm requires a replay buffer containing tuples of the global state and joint action (s, a). For a single EV agent, its contribution to this tuple would include its private local state (e.g., current battery SoC, precise location, intended departure time) and its chosen action (charging rate). This means that every participating EV owner would be required to continuously stream their private data to the operator's central server. Such a requirement is untenable; it reveals sensitive lifestyle patterns, such as when a user is home or at work, and creates a massive repository of private data that is a prime target for misuse or attack. This architectural requirement of standard CTDE creates a direct and irreconcilable conflict with the data sovereignty

principles essential for public participation.

Recent literature has attempted to reconcile this conflict through various privacy-preserving mechanisms, primarily categorized into Federated Reinforcement Learning, cryptographic methods, and differential privacy. However, each approach introduces significant trade-offs that limit their applicability to real-time, high-fidelity DT coordination.

Cheruiyot et al. [23] and Yang et al. [141] explore Federated Reinforcement Learning (FRL), where agents collaboratively train a model by sharing gradients rather than raw data. While this protects local observations, it focuses on learning shared policy parameters rather than coordinating specific real-time actions. Furthermore, as noted in Section 3.3.2, these methods often result in opaque models that lack the explainability required for critical infrastructure.

To enable secure value decomposition, several studies have applied cryptographic techniques. Mukherjee et al. [91] proposed a Secure Multi-Party Computation (MPC) framework for MARL in supply chains. Similarly, Gohari et al. [35] introduced Privacy-Engineered Value Decomposition Networks (PE-VDN) using MPC and differential privacy, while Yuan et al. [146] employed homomorphic encryption to protect critic updates. However, these cryptographic operations introduce significant computational overhead. Mukherjee et al. [91] explicitly notes the "programmatic intractability" of implementing policy gradients with such heavy protocols, making them ill-suited for the high-frequency control loops of energy grids.

Another stream of research focuses on obfuscating data. Zhao et al. [152] proposed the Differentially Private Multi-Agent Communication (DPMAC) algorithm, which adds noise to communication channels. Lin et al. [75] quantified the trade-off inherent in this approach: increasing privacy protection (noise) directly degrades the convergence and utility of the learned policy. Alternatively, Ahmed and Ghasemi [1] and Ye et al. [142] proposed restricting the *type* of information shared, such as exchanging only trajectory rankings or embedded representations rather than raw states. While these methods reduce data leakage, they often limit the richness of the coordination signal, preventing the

sophisticated "what-if" negotiation that DTs are capable of.

Critical Analysis: Current approaches to privacy-preserving MARL present a "trilemma" between **performance**, **privacy**, and **computational feasibility**. Cryptographic methods like MPC provide strong privacy but are computationally prohibitive for real-time control loops. Differential privacy offers a tunable trade-off but inherently degrades the utility of the learned policy by injecting noise. Federated Learning addresses data locality but fails to support the granular, action-level coordination required for physical systems. Consequently, a significant gap remains for a framework that can leverage the specific capabilities of DTs, namely simulation, to generate high-fidelity coordination signals that are both privacy-preserving and computationally efficient.

3.6 Synthesis and Identification of the Research Gap

To synthesise the analysis presented in the preceding sections, Table 3.1 provides a comparative summary of the reviewed approaches, highlighting their architectural characteristics, privacy mechanisms, and specific limitations in the context of DT coordination.

The review of the state-of-the-art reveals a significant three-way disconnection at the intersection of DT architectures, collaborative frameworks, and intelligent control methodologies. This observation is supported by recent comprehensive surveys; for instance, Sai et al. [110] explicitly identify the development of secure, privacy-preserving coordination mechanisms as a critical open challenge for the next generation of Digital Twin Networks.

1. **Architectural Gap:** While decentralised DT architectures effectively address the scalability and privacy limitations of centralised systems, they currently lack the mechanisms for sophisticated, learning-based coordination. Existing decentralised frameworks often rely on static rule-based logic or manual consensus, failing to capture the emergent optimisation capabilities of centralised counterparts.

Table 3.1: Comparison of State-of-the-Art Approaches in Digital Twin Coordination.

Approach	Architecture	Privacy Mechanism	Coordination Method	Primary Limitation
Centralised DTs [4, 20]	Cloud-Centric	None (Raw Data Aggregation)	Global Optimisation	Privacy violation & Single point of failure.
Federated Learning [111, 113, 104]	Edge / Distributed	Gradient Sharing	Shared Model Training	Opaque "black-box" models; lacks real-time action coordination.
Blockchain / P2P [109, 33, 70]	Decentralised	Cryptography / Ledger	Smart Contracts / Consensus	High latency; computationally expensive for real-time control.
Conventional Optimisation [151, 153, 143]	Centralised or Distributed	Model-based	Mathematical Programming	Brittle; relies on static models; computationally heavy.
Standard MARL (CTDE) [79, 105, 144]	Centralised Training	None (Centralised Critic)	Learned Policy (RL)	Privacy violation (requires global state access).
Privacy-Preserving MARL [91, 35, 152]	Federated or Encrypted	Noise Injection / MPC	Learned Policy (RL)	Utility loss (noise) or computational intractability.
Proposed Framework	**Decentralised P2P**	**Decision-Level Sharing**	**Simulation-Assisted RL**	**Discussed in 7.5.**

2. **Algorithmic Gap:** Conversely, state-of-the-art MARL algorithms, such as MADDPG, offer powerful coordination capabilities but are architecturally incompatible with privacy-sensitive deployments. They rely on the CTDE paradigm, which assumes a centralised critic has complete access to the raw private states of all agents.

3. **Methodological Gap:** Alternative privacy-preserving collaboration methods, such as FL, typically produce opaque black-box models. These lack the explainability and simulation-based "what-if" analysis capabilities that are fundamental to the definition and utility of a DT.

Consequently, a critical research gap exists for an integrated framework that can: (1) architecturally support peer-to-peer collaboration without exposing raw data; (2) mechanically ensure trust in an open environment where peers might be unreliable; and (3) algorithmically enable agents to learn coordinated policies using only privacy-preserving, high-level information.

3.7 Thesis Objectives and Contributions

To address the identified gaps, this thesis pursues the following specific research objectives:

- **Objective 1: Design a Decentralised "Decision-Level" Framework.** To address the **Architectural Gap** by designing a middleware that enables coordination without centralisation. This objective aims to overcome the limitations of existing decentralised architectures, which lack mechanisms for sophisticated collaboration, by implementing a protocol for sharing high-level decisions rather than raw data.
- **Objective 2: Develop a Trust Mechanism for Open Systems.** To address the reliability gap in open P2P networks. Existing reputation systems rely on subjective consensus which is vulnerable in DTNs. This objective aims to fill this gap by developing a verification mechanism that anchors trust in objective physical reality, ensuring robustness against unreliable peers.
- **Objective 3: Propose a Privacy-Preserving MARL Algorithm.** To address the **Algorithmic Gap** where standard MARL violates privacy. This objective aims to resolve the conflict between CTDE requirements and data sovereignty by introducing **DT-MADDPG**, which utilises a collaborative simulation model to replace the need for raw data aggregation during training.
- **Objective 4: Validate via Large-Scale V2G Simulation.** To address the need for empirical validation of these novel privacy-preserving mechanisms. This objective aims to demonstrate that the proposed framework can bridge the identified performance gap, achieving grid stability comparable to centralised baselines while maintaining the privacy benefits of decentralisation.

3.8 Summary

This chapter has critically analysed the landscape of DTs, collaborative frameworks, and multi-agent coordination. The review established that while the individual components for decentralised manage-

ment exist, there is no cohesive solution that simultaneously satisfies the requirements of scalability, privacy, and intelligent optimisation. Specifically, current architectures force a trade-off between the privacy of decentralisation and the performance of centralised control. The literature indicates a clear need for a new architectural paradigm that leverages the predictive capabilities of DTs to enable trusted, decision-level collaboration.

With the research gaps identified and the objectives defined, the remainder of this thesis is dedicated to constructing this solution. The following chapter, **Chapter 4: Framework Design**, addresses the first objective by detailing the architectural design of a fully decentralised DT network that enables privacy-preserving information exchange.

Chapter 4

Framework Design: A Decentralised Architecture for Digital Twin Cooperation

4.1 Introduction

The literature review in Chapter 3 highlighted a critical contrast in the management of Large Complex Systems. On one hand, centralised Digital Twin architectures offer global optimisation but suffer from scalability bottlenecks, single points of failure, and, most critically, privacy violations in multi-stakeholder environments. On the other hand, existing decentralised approaches often focus on data interoperability without providing a mechanism for intelligent, privacy-preserving collaboration.

To address these limitations, this chapter presents the design of a novel, fully decentralised Digital Twin framework. This framework serves as the structural foundation for "Cooperation", defined in this thesis as the capability and willingness of autonomous agents to exchange information to benefit the collective.

The core philosophy of this design is a paradigm shift from *Data-Level Integration* to *Decision-Level*

Cooperation. Traditional frameworks rely on the aggregation of raw sensor data (e.g., precise GPS coordinates or user usage habits) to a central node. In contrast, the proposed framework enables agents to process raw data locally and exchange only high-level *decision information* (e.g., predicted energy demand or intended charging schedules). This approach maintains the privacy of sensitive underlying data while still providing the network with the necessary information to achieve system-wide objectives. The core philosophy of this design is a paradigm shift from *Data-Level Integration* to *Decision-Level Cooperation*. Traditional frameworks rely on the aggregation of raw sensor data (e.g., precise GPS coordinates or user usage habits) to a central node. In contrast, the proposed framework enables agents to process raw data locally and exchange only the **Minimum Viable Information (MVI)** in the form of high-level *decision information* (e.g., predicted energy demand or intended charging schedules). This approach maintains the privacy of sensitive underlying data while still providing the network with the necessary information to achieve system-wide objectives.

This chapter is organised as follows: Section 4.2 defines the key design requirements derived from the challenges of Vehicle-to-Grid (V2G) systems. Section 4.3 details the high-level network topology and the internal architecture of the proposed Digital Twin nodes. Section 4.4 describes the specific mechanism and protocols for decision-level information sharing. Section 4.7 analyses the privacy-preservation characteristics of the design. Finally, Section 4.8 outlines the prototype implementation used to validate the framework.

The proposed architecture forms a **fully decentralised, peer-to-peer federation** of autonomous DT agents. Each agent processes sensitive data locally and collaborates by sharing only high-level, abstracted 'decision information' such as predicted power demand rather than raw sensor data. This design ensures privacy and scalability by distributing intelligence to the network edge and minimising data transmission, forming the foundation for trusted and intelligent coordination. (Note that while the operational execution and communication layers defined in this chapter are strictly decentralised, the MARL training mechanism introduced later in Chapter 6 temporarily leverages a hybrid

approach to construct a global model for offline learning).

4.2 Design Requirements

To overcome the limitations of centralised architectures and address the specific challenges of the V2G application domain, the proposed DT framework is designed according to four primary requirements: Privacy Preservation, Scalability, Local Autonomy, and Interoperability.

4.2.1 Privacy Preservation

The primary constraint in multi-owner systems is the protection of sensitive data. In a V2G context, raw data such as an EV's Global Positioning System (GPS) location, the owner's driving schedule, and battery usage habits can reveal intimate details about a user's life.

Requirement: The framework must ensure that raw, sensitive data remains exclusively at the local node (the "Edge"). Information shared with the network must be abstracted to a level that prevents the reconstruction of the user's private state or habits by peers or malicious observers.

4.2.2 Scalability and Robustness

Large Complex Systems, such as national power grids, involve thousands to millions of interacting components. Centralised architectures often encounter bandwidth bottlenecks and processing latency when handling real-time data from such a vast number of endpoints. Furthermore, reliance on a central server creates a single point of failure.

Requirement: The architecture must be fully distributed. It should support the dynamic addition or removal of nodes without significant reconfiguration or performance degradation. The failure of any single node (or group of nodes) must not compromise the integrity of the broader network.

4.2.3 Local Autonomy

In a decentralised environment, agents are often self-interested and owned by different stakeholders. A central controller cannot simply "command" a private vehicle to discharge its battery if doing so conflicts with the owner's immediate travel needs.

Requirement: The framework must respect the sovereignty of individual agents. Each DT must retain the ultimate authority to make decisions based on its local objectives and constraints. "Cooperation" must be voluntary and driven by incentive or mutual benefit, rather than coerced by a central algorithm.

4.2.4 Interoperability

A realistic DT network is heterogeneous; it is composed of assets from different manufacturers with varying capacities, dynamics, and control interfaces (e.g., different brands of EVs or battery storage systems).

Requirement: The framework must provide a standardised communication interface and protocol. This allows heterogeneous agents to exchange decision-level information and coordinate actions regardless of their underlying physical hardware or internal model structure.

4.2.5 Justification and Completeness of Requirements

The four requirements presented: Privacy Preservation, Scalability, Local Autonomy, and Interoperability, are asserted to be both sufficient and complete for the architectural design phase of this framework. Their justification stems directly from the core research problem and the gaps identified in the literature review.

Sufficiency: The requirements are considered sufficient because, when satisfied, they directly address the fundamental "Privacy-Coordination Paradox."

- **Privacy Preservation** and **Local Autonomy** are direct responses to the "Willingness to Cooperate" challenge, ensuring that self-interested, private stakeholders have the guarantees needed to participate.
- **Scalability and Robustness** address the technical failures of centralised architectures when applied to Large Complex Systems.
- **Interoperability** is a pragmatic necessity for any real-world deployment, which will inevitably involve heterogeneous assets.

Together, these four pillars form a foundation upon which a cooperative system can be built without resorting to a centralised, privacy-invasive model.

Distinction: It is crucial to distinguish between Privacy Preservation and Local Autonomy, as they address different facets of stakeholder sovereignty. **Privacy Preservation** is an *information-centric* requirement, concerned with protecting an agent's internal state from being exposed or inferred. It answers the question: "What can others know about me?". In contrast, **Local Autonomy** is a *control-centric* requirement, guaranteeing an agent's right to make its own decisions based on its local objectives. It answers the question: "What can others make me do?". While related, they are not interchangeable. A system could preserve privacy by using encrypted commands from a central controller (violating autonomy), or it could grant full autonomy while requiring agents to broadcast all their data to coordinate (violating privacy). This framework mandates both, ensuring agents have sovereignty over their data and their actions.

Completeness: While other requirements such as security against specific attacks, real-time performance guarantees, or economic fairness are critical, they are considered second-order concerns that build upon this architectural foundation. For instance, the reliability of information is not a primary architectural requirement here but is addressed by the **Trust Mechanism** in Chapter 5. Similarly, the optimisation of coordination is handled by the **Intelligent Coordination** algorithm in Chapter 6.

Therefore, these four requirements are considered complete for the specific purpose of defining the high-level system structure, which is the primary goal of this chapter.

4.3 High-Level System Architecture

To satisfy the requirements outlined above, the proposed framework adopts a fully decentralised, P2P architecture. By structuring the network as a **fully decentralised federation** of twins, this design eliminates the reliance on a central server or an upper-level global DT at runtime, distributing both the computational load and the decision-making authority across the network edges.

4.3.1 Network Topology

The system is modelled as a connected graph $G = (V, E)$, where V represents the set of DT nodes and E represents the communication links between them.

Unlike hierarchical models where edge nodes communicate only with a parent controller, this framework utilises a flat P2P topology. Nodes discover and establish connections with neighbors based on:

- **Physical Proximity:** Agents that are geographically close (e.g., EVs at the same charging station).
- **Electrical Connectivity:** Agents connected to the same grid feeder or transformer, ensuring that coordination addresses local grid constraints.
- **Logical Relevance:** Agents with similar operational goals or complementary capabilities.

This topology ensures that communication is localised, reducing network latency and ensuring that coordination efforts are focused where they are most physically relevant.

4.3.2 Peer Discovery Mechanism

In a dynamic P2P network where nodes join and leave asynchronously, a robust peer discovery mechanism is critical. The framework employs a two-phase process that combines a reliable bootstrapping method with a scalable gossip protocol for continuous topology maintenance.

Phase 1: Initial Bootstrap

When a new DT agent comes online (e.g., an EV plugs into a charging station), it faces the "cold start" problem of having no knowledge of other active peers. To solve this, the agent contacts one of a small, pre-configured list of **bootstrap nodes**. These are stable, well-known entry points to the network, typically maintained by the system operator for high availability. To ensure robustness, the agent's configuration contains multiple bootstrap addresses.

Upon receiving a connection request, the bootstrap node returns a small, random subset of currently active peers from its cache, along with their relevant metadata (e.g., virtual IP address, grid feeder ID, geographical zone). This initial list provides the new agent with its first set of neighbours, enabling it to join the network without relying on a centralised registry.

Phase 2: Dynamic Discovery via Gossip Protocol

Once bootstrapped, the agent transitions to a **gossip-based protocol** to dynamically discover new peers and maintain an up-to-date view of the network topology. This framework employs a *push-pull* gossip mechanism for efficient information propagation:

- **Gossip Cycle:** Periodically (e.g., every 30 seconds), each agent initiates a gossip cycle. It randomly selects a small number of its known neighbours (its "fanout") and sends them a *gossip message*.
- **Message Structure:** A gossip message contains a list of peer entries known to the sender. Each

entry is a tuple of *(peer_id, metadata, timestamp)*. The *timestamp* indicates the last time the peer was known to be active.

- **State Reconciliation:** Upon receiving a gossip message, a peer merges the incoming list with its own local peer list. It updates its entry for a peer if the incoming timestamp is more recent and adds any new peers it was unaware of. Stale entries (those with old timestamps) are periodically pruned to keep the list fresh and bounded in size. This push-pull exchange ensures that information about new nodes and node departures spreads exponentially fast throughout the network.
- **Relevance Filtering:** The agent's *Scope Manager* (as detailed in Chapter 5) uses the metadata in the received peer list to filter and establish persistent connections only with relevant neighbours (e.g., those on the same grid feeder). This forms the localised sub-graphs necessary for efficient, targeted communication.

Connection Liveness and Fault Tolerance

The gossip protocol itself provides a passive, low-overhead liveness detection mechanism. A node that goes offline will stop participating in gossip exchanges. Consequently, the timestamp associated with it in other peers' lists will become stale, and the entry will eventually be pruned. This is more efficient than requiring active, high-frequency "heartbeat" messages between all connected peers.

If a node becomes partitioned from the network and loses all its neighbours, it can detect this state after several failed gossip cycles. It then automatically reverts to Phase 1, re-contacting a bootstrap node to rejoin the network, making the system self-healing.

Practical Implementation with ZeroTier

The abstract mechanisms described above map cleanly onto concrete network virtualisation technologies like **ZeroTier**.

- **Bootstrap:** ZeroTier's public "root servers" (or privately hosted "moons") can serve as the bootstrap nodes. They handle the initial peer discovery and, crucially, the Network Address Translation (NAT) traversal required for devices behind firewalls to connect directly.
- **Virtual LAN:** Once an agent joins a ZeroTier network, it is assigned a virtual IP address and becomes part of a secure, encrypted virtual LAN. All communication within this LAN is peer-to-peer by default.
- **Gossip:** The gossip protocol can then be implemented efficiently using simple UDP broadcasts or multicasts over this virtual LAN, as if all nodes were on the same physical switch. This greatly simplifies the implementation of the discovery and communication layers.

This hybrid approach ensures both robust network entry and scalable, localised discovery, preventing the need for a centralised registry and aligning with the core requirements of decentralisation and robustness.

4.3.3 Node Architecture: The Digital Twin Agent

Each node in the network is an autonomous agent, which is a DT representing a specific physical asset (e.g., an EV). The internal architecture of a single node is composed of four distinct modules:

1. **Database (The Interface):** This module is responsible for the continuous acquisition of raw data from the physical asset (e.g., battery State of Charge, current location). It acts as the local "Digital Shadow," ensuring the node's internal state reflects physical reality. Crucially, this raw data *never* leaves this module to be shared externally.
2. **Prediction Engine (The Simulator):** This module contains the high-fidelity physics or data-driven model of the asset. It enables the DT to perform "what-if" simulations, predicting future states (e.g., "If I charge at 5kW for 1 hour, my battery will be at 80%"). This simulation capability is the engine of the "Predictive" twin. It may also utilise AI-related models to forecast

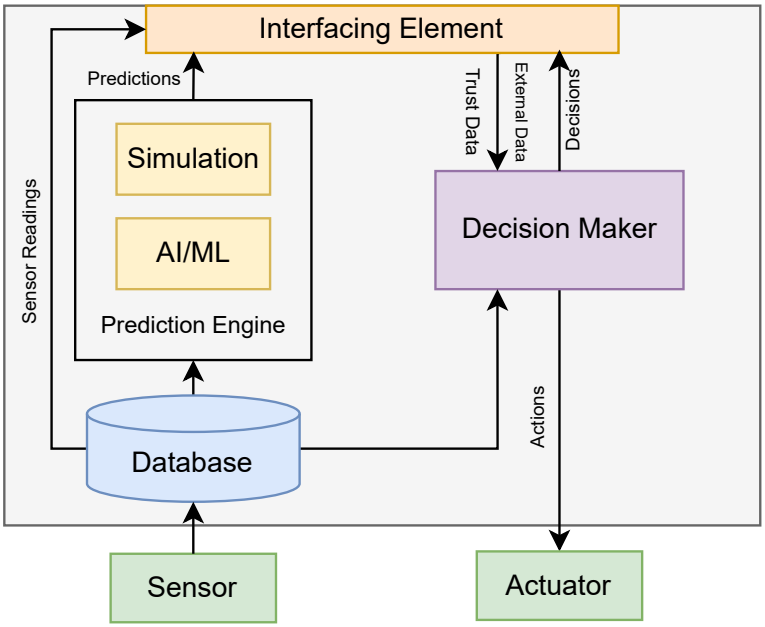

Figure 4.1: Reference Architecture of a Digital Twin Node in the Proposed Framework.

external conditions (e.g., electricity prices, grid load).

3. **Decision Maker (The Brain):** This module hosts the decision-making algorithm (specifically, the MARL agent discussed in Chapter 6). It takes inputs from the Prediction Engine (predictions) and the Interfacing Element (neighbor information) to determine the optimal action.

4. **Interfacing Element (The API Gateway):** This module represents the minimal modifications required within an existing DT to enable collaboration. It provides standardised API interfaces and is responsible for:

    - Extracting and formatting internal state data and intentions into abstract "Decision Information" (I_{dec}), strictly adhering to an *a priori* agreed-upon ontology of "Interfacing States" (further detailed in Chapter 5).
    - Transmitting this formatted data to the external Middleware component (detailed in Chapter 5) which runs alongside the DT.
    - Receiving aggregated peer information (the "Context Vector") from the Middleware and

passing it to the Decision Maker.

4.3.4 The Digital Twin-Middleware API

A critical aspect of this architecture is the formal interface, or API, between the Digital Twin process and the Middleware process. This API is designed as a clean, local interface that decouples the DT's internal application logic from the complexities of peer-to-peer networking, which are handled entirely by the Middleware. The interaction is based on two primary function calls.

1. **`send_intent(I_dec):`** This function is called by the DT's Interfacing Element to push its locally generated decision information to the Middleware. The payload, `I_dec`, is a data structure containing the Prediction and Decision Tuples defined in Chapter 5. Upon receiving this call, the Middleware is responsible for serialising the data (e.g., to JSON) and broadcasting it to the relevant peers over the P2P network.

2. **`on_receive_context(C_i):`** This is a callback function that the DT must implement and register with the Middleware. The Middleware's Verifier and History Database continuously listen for broadcasts from peers. After receiving, verifying, and aggregating peer intentions into a "Context Vector" `C_i` (using the trust-weighted mechanism from Chapter 5), the Middleware calls this function to push the aggregated context back to the DT. The DT's Decision Maker then uses this vector as an input for its next decision cycle.

This API design enforces a clean separation of concerns. The Digital Twin developer can focus entirely on the high-fidelity modelling and decision-making logic of the physical asset, treating the Middleware as a "black box" for decentralised communication. The following pseudo-code illustrates the interaction:

```
// Inside the Digital Twin's Interfacing Element
function execute_decision_cycle():
```

```
    // 1. Generate local intent based on internal state
    intent_payload = self.prediction_engine.generate_intent()

    // 2. Push intent to the middleware via the local API
    middleware_api.send_intent(intent_payload)

// Callback function implemented by the DT
function on_receive_context(context_vector):
    // 3. Use aggregated peer data for the next decision
    self.decision_maker.update_context(context_vector)

// ---

// Inside the Middleware Process
function handle_peer_broadcast(peer_message):
    // 1. Receive and verify peer message
    self.verifier.process(peer_message)

    // 2. Aggregate into a new context vector
    new_context = self.aggregator.get_context()

    // 3. Push context to the local DT via the callback
    digital_twin_api.on_receive_context(new_context)
```

4.4 The Decision-Level Information Sharing Mechanism

The fundamental enabler of cooperation in this framework is the mechanism by which agents exchange information. Standard data-driven approaches often require agents to share raw observations to build a central model. This framework employs a *Decision-Level Sharing* mechanism, which decouples the internal state of the agent from its external communication.

4.4.1 Information Abstraction

We formally distinguish between two categories of information within a Digital Twin node:

- **Raw Data (D_{raw}):** This includes high-fidelity sensor readings and sensitive user constraints. In the V2G context, this comprises the vehicle's exact GPS location, the user's specific departure time, the current battery SoC, and historical usage patterns.
- **Decision Information (I_{dec}) serving as the Minimum Viable Information (The "Intent"):** This represents the *outcome* of the local agent's processing. As the practical implementation of the MVI concept introduced in Chapter 1, it is a projection of the agent's future **intent** or predicted impact on the environment. Formally, *intent* maps directly to **collaboration** by defining the structured format of what agents communicate (e.g., "I plan to draw 5kW"), and serves as the primary input for **coordination**, where these interacting intents are algorithmically resolved (e.g., via the MARL agent in Chapter 6). Crucially, what falls *outside* the scope of intent are the underlying justifications, private constraints (e.g., "I am charging because my owner leaves at 8 AM"), and the raw physical states (e.g., battery degradation metrics) that formulated the decision. To ensure standardised communication across the network, this abstract information is technically structured and transmitted as **Prediction Tuples** and **Decision Tuples**, which are formally defined in Chapter 5.

In the proposed framework, D_{raw} remains strictly local. The agent processes D_{raw} using its Local

Model and Decision Maker to generate I_{dec}, which is then passed to the Middleware to be broadcast to neighbors. Formally, if π is the agent's policy and M is its internal model:

$$I_{dec} = M(\pi(D_{raw}))$$

For a VPP or V2G agent, I_{dec} typically takes the form of a **Power Profile**: a vector indicating the intended power import (charging) or export (discharging) for the next N time-steps (e.g., $[+5kW$, $+5kW$, $-2kW$, $\dots]$).

For example, consider an EV owner who arrives home at 6 PM and needs their car fully charged by 8 AM the next morning. The raw data (D_{raw}) includes the current State of Charge (e.g., 20%), the departure time, and the target SoC (e.g., 90%). The EV's DT processes this private information and, considering real-time electricity prices, formulates an optimal charging schedule. Instead of broadcasting its private constraints, it shares only the resulting decision information (I_{dec}): a power profile indicating its planned energy draw over the next 14 hours (e.g., "draw 0kW from 6-11 PM, draw 7kW from 11 PM-7 AM"). This allows other agents on the local grid to anticipate this load without ever knowing the user's personal schedule or battery status.

4.4.2 The Sharing Protocol

The cooperation follows an iterative protocol at each decision time-step t. This section details the steps of the protocol and its temporal dynamics.

Protocol Steps

At each time-step, every agent performs the following sequence of actions:

1. **Observation:** Each agent i observes its local state s_i (derived from D_{raw}).
2. **Local Optimisation:** The agent calculates a tentative action a_i based on its current policy.

3. **Prediction:** The agent uses its internal simulation to predict the impact of a_i over the near future horizon, generating the intention profile I^i_{dec}.
4. **Broadcast:** The agent broadcasts I^i_{dec} to its set of connected neighbours $\mathcal{N}_i$.
5. **Aggregation:** The agent receives intention profiles from its neighbours via the Middleware. It aggregates these (e.g., summing the predicted power loads) to form a "Context Vector" C_i, which represents the collective state of the local grid for the next time-step.

Temporal Dynamics: Logical Synchronicity

While the underlying network is physically asynchronous, the protocol itself operates under a **logically synchronous** model. This means that cooperation is structured into discrete, iterative rounds, where each round corresponds to a decision time-step (e.g., every 15 minutes for grid balancing).

This design choice does not imply a requirement for strict, low-level clock synchronisation across all nodes, which would be impractical in a large-scale P2P network. Instead, it defines a high-level protocol timing. Agents are expected to broadcast their intentions for the upcoming time-step $t+1$ within a flexible time window during the current step t. An agent's Decision Maker then uses the most recently received "Context Vector" to make its final decision for that round. This approach provides the structural benefits of a round-based system, making the coordination problem more tractable while retaining the robustness to handle network latency and asynchronous message arrival inherent in a real-world decentralised system. If a message from a peer is delayed, the agent can proceed using the last known information, with the Trust Mechanism (Chapter 5) eventually down-weighting consistently latent or unavailable peers.

Conflict Detection and Resolution

A key function of sharing decision-level information is to enable proactive conflict detection. A conflict arises when the aggregated intentions of neighboring agents, represented by the Context Vector

C_i, lead to a predicted violation of a physical or logical constraint. For example, in a V2G network, a conflict occurs if the sum of predicted power demands from all EVs on a local feeder exceeds the transformer's maximum capacity.

The resolution of such conflicts is not handled through direct, explicit negotiation between agents in this architectural layer. Instead, the framework facilitates an **iterative adaptation** process. Upon detecting a potential conflict from the received Context Vector, each agent's Decision Maker is responsible for re-evaluating and adjusting its own plan. For instance, an agent might choose to defer some of its planned charging load to a later time to alleviate the predicted grid stress. This adjusted plan results in a new, less conflicting intention profile (I'_{dec}) being broadcast in the subsequent round. As other agents receive this updated information, they too will adjust their plans, leading to a dynamic, decentralised process that converges towards a mutually feasible state. While simple agents might use heuristic rules for this adjustment (e.g., "reduce demand by 10% if a conflict is detected"), the core of this thesis is to enable *intelligent* resolution. The MARL algorithm detailed in Chapter 6 learns a sophisticated policy that proactively avoids or optimally resolves these conflicts to balance individual goals with global system stability.

4.4.3 Types of Alignment Supported by Intent

The exchange of decision-level intent (I_{dec}) serves as the primary vehicle for achieving system-wide "alignment" across three distinct dimensions within the decentralised network:

1. **Semantic Alignment (Collaborative Interoperability):** Heterogeneous assets (e.g., an EV and a solar panel) possess fundamentally different internal physics and state spaces (D_{raw}). The framework achieves semantic alignment by forcing these disparate agents to project their internal states into a standardised, mutually comprehensible language of "intent" (e.g., future power profiles), allowing them to collaborate without requiring a unified global physics model.

2. **Temporal Alignment (Action Deconfliction):** By broadcasting future intentions rather than

just current states, agents can foresee the aggregate physical impact of their planned actions. This temporal foresight allows the network to align physical actions in time, proactively avoiding constraint violations (such as simultaneous charging causing a transformer overload) through the iterative adaptation process described above.

3. **Objective Alignment (Reconciling Utilities):** The most challenging dimension is aligning an agent's private utility (e.g., the financial cost of charging) with the global welfare of the system (e.g., grid stability). It is crucial to note that while "intent" communicates *what* an agent will do, it does not inherently force an agent to act selflessly. The framework supports objective alignment by using the shared intents as the environmental context (the "Context Vector") required for intelligent agents to learn optimal policies. The actual mathematical reconciliation of these utilities is achieved via the hierarchical reward structures and the MARL algorithm detailed in Chapter 6, which guides the agents to formulate intents that are both locally beneficial and globally responsible.

4.4.4 The Principle of Willingness

This mechanism explicitly addresses the "Willingness to Cooperate" challenge identified in the introduction. In multi-owner systems, willingness is often a function of privacy risk. By ensuring that the broadcast information I_{dec} (e.g., "I need 5kW") is abstract and divorced from the sensitive context (e.g., "I am at 123 Main St and need to leave at 5 PM"), the framework lowers the barrier to entry. This architectural guarantee fosters a higher degree of participation (willingness) from private asset owners compared to systems that demand full data transparency.

4.5 Mathematical Formulation: The Framework as a Dec-POMDP

To mathematically formalise the proposed architecture and explicitly link the privacy constraints to the coordination mechanism, the entire decentralised Digital Twin network is modelled as a Decentralised Partially Observable Markov Decision Process (Dec-POMDP) [15, 95].

4.5.1 Justification for Partial Observability

The choice to model the system as a Dec-POMDP rather than a standard fully observable Markov Decision Process (MDP) is a direct mathematical consequence of the "Privacy-Coordination Paradox" introduced in Chapter 1.

In a standard, fully observable formulation, an agent (or a central controller) bases its decisions on the complete global state of the environment ($\mathscr{S}$). This would necessitate real-time access to the exact, raw data (D_{raw}) of every single agent in the network. While this full observability allows for perfect mathematical optimisation, it completely violates the data sovereignty requirement, triggering the "Optimisation-Driven Exposure" failure mode.

The proposed framework enforces strict data sovereignty. Therefore, the true global state of the system is deliberately hidden from individual agents, making the environment inherently **partially observable**. An agent cannot observe the private parameters of its peers; it relies solely on its local, privacy-preserving observations. The Dec-POMDP framework rigorously captures this paradigm, mathematically modelling how decentralised agents can collaborate despite operating strictly under local, partial views of the system.

4.5.2 Generic Formal Definition

Formally, the framework is defined by the tuple $\langle N, \mathscr{S}, \mathscr{A}, \mathscr{P}, \mathscr{R}, \Omega, \mathscr{O}, \gamma \rangle$:

- $N = \{1, \ldots, n\}$ is the set of autonomous Digital Twin agents.

- $\mathscr{S}$ is the set of global system states (the true, hidden state of the entire network, equivalent to the union of all agents' D_{raw} and the broader environment state).
- $\mathscr{A} = \{A_1, \ldots, A_n\}$ represents the set of joint actions, where $a_i \in A_i$ is the physical action taken by the local Decision Maker.
- $\mathscr{P}$ denotes the state transition probability $\mathscr{P}(s'|s,a)$, representing the physical dynamics of the environment.
- $\Omega = \{\Omega_1, \ldots, \Omega_n\}$ is the set of observation spaces.
- $\mathscr{O}$ is the observation function $\mathscr{O}(o|s,a)$. In this framework, the observation $o_i \in \Omega_i$ is explicitly constructed by the Interfacing Element as a concatenated vector:
  - **Private Local State:** The agent's own internal D_{raw} (e.g., local constraints and goals).
  - **Aggregated Peer Information:** The "Context Vector" C_i, constructed by aggregating the Decision Information (I_{dec}) shared by neighbors via the Middleware.
- $\mathscr{R}$ is the reward function $\mathscr{R}(s,a)$. In a multi-owner system, this is typically a hybrid function balancing global system health with individual local utility.
- $\gamma \in [0,1)$ is the discount factor, determining the importance of future rewards.

By defining the architecture through this tuple, we provide a rigorous mathematical foundation for the subsequent chapters. Chapter 5 focuses on ensuring the reliability of the observed Context Vector C_i via trust mechanisms, while Chapter 6 introduces the specific reward functions $\mathscr{R}$ and the learning algorithms required to solve this Dec-POMDP.

4.6 Communication Complexity Analysis

The scalability of a distributed system is fundamentally constrained by its communication complexity. In the proposed framework, communication occurs primarily through two distinct processes: the ongoing peer discovery via the gossip protocol and the iterative exchange of decision information.

4.6.1 Peer Discovery (Gossip Protocol)

The gossip protocol is used for network maintenance and topology discovery. Its complexity can be analysed as follows:

- **Message Size:** Each gossip message contains a list of known peers. If the list size is bounded by a constant k, the message size is $O(k)$.
- **Message Count:** In each gossip round, an agent sends a message to a small, constant number of neighbors (its fanout, f). For a network of N agents, the total number of messages per round is $O(N \cdot f)$, which simplifies to $O(N)$ as f is a constant.
- **Overall Complexity:** The gossip protocol is low-frequency (e.g., every 30-60 seconds) and lightweight. Its primary cost is a constant background traffic of $O(N)$ messages per round, ensuring the network view converges in $O(\log N)$ rounds. This is a highly scalable mechanism for maintaining network awareness.

4.6.2 Decision Information Exchange

This is the primary communication pattern for coordination.

- **Message Size:** Each message contains a Decision Information packet, which is a power profile over a future horizon H. The size of this packet is therefore $O(H)$.
- **Message Count:** At each decision time-step, an agent multicasts its packet to its set of relevant

peers. The size of this peer set is determined by the Scope Manager and is denoted by p. In a well-structured network with locality (e.g., based on grid feeders), p is expected to be much smaller than N ($p \ll N$). The total number of messages generated across the network per time-step is $O(N \cdot p)$.

- **Overall Complexity:** The total communication volume per time-step is $O(N \cdot p \cdot H)$. Since H is a fixed parameter and p is a small, managed constant representing local connectivity, the communication volume scales linearly with the number of agents, $O(N)$.

4.6.3 Comparison with Centralised Architectures

The linear scaling $O(N)$ might seem similar to that of a centralised system, where N agents each send one message to a central server. However, the critical difference lies in the distribution of the communication load.

- **Centralised Bottleneck:** In a centralised model, the server becomes a bottleneck as it must receive and process N incoming messages in each time-step. The load on this single node is $O(N)$.
- **Decentralised Load Distribution:** In the proposed P2P framework, the communication load is distributed. The number of messages any single node must receive and process in a time-step is bounded by the size of its peer set, p. As p is managed to be small and constant, the load on any individual node is $O(p)$, which is $O(1)$.

This distribution of load is the key to the framework's scalability and robustness. It avoids the single point of failure and the processing bottleneck inherent in centralised designs.

4.6.4 Mitigation Strategies

The iterative nature of conflict resolution can lead to high-frequency message exchange. To manage this, the framework employs two key mitigation strategies:

- **Threshold-Based Broadcasting:** Agents only broadcast an updated intention profile if the change from the previous one exceeds a significance threshold.
- **Rate Limiting:** A minimum time interval is enforced between consecutive broadcasts from a single agent.

These strategies ensure that the communication overhead remains manageable even in volatile conditions, balancing the need for rapid adaptation with network scalability. The quantitative impact of these factors is evaluated in the scalability analysis in Chapter 5.

4.7 Privacy Preservation Strategy

The architectural design of the proposed framework inherently prioritises privacy through the principle of data minimisation and local execution. This section analyses how the decision-level sharing mechanism acts as a robust privacy barrier.

4.7.1 Privacy through Data Abstraction

The core privacy mechanism is the decoupling of the *decision* from the *observation*. In a traditional centralised system, the mapping function $f : S \rightarrow A$ (where S is state and A is action) is executed on a remote server, requiring S to be transmitted. In the proposed framework, f is executed locally.

Consequently, the information leaving the node is an abstraction. For example, a broadcast intention of "Demand: +7kW" is highly ambiguous regarding the underlying state. It could result from:

- A low battery (SoC 20%) charging urgently for a trip in 1 hour.

- A half-full battery (SoC 50%) charging slowly for a trip in 5 hours.
- A stationary storage unit responding to a price signal.

Because multiple internal states (s_1, s_2, s_3) can map to the same decision output (d), it becomes computationally infeasible for an external observer (a peer or eavesdropper) to inverse the function and reconstruct the sensitive raw state S with high confidence.

Resilience to Suboptimal State Mapping

It is important to acknowledge that the degree of privacy preserved relies heavily on the system designer's selection of what constitutes "Decision Information." If a system is poorly designed such that internal states are mapped one-to-one with broadcast intentions (e.g., an agent directly broadcasts its precise SoC curve instead of an aggregated power profile), the privacy-preserving benefits of data abstraction are significantly degraded. However, even in such a worst-case scenario, the proposed framework remains highly useful. By distributing the architecture, the framework inherently eliminates the central computational bottleneck and the single point of failure associated with cloud-centric monolithic twins. Furthermore, it still enables the application of the Trust Mechanism (Chapter 5) to verify peer reliability and supports highly scalable, decentralised coordination algorithms (Chapter 6) without relying on a central orchestrator. Thus, while optimal privacy requires careful state abstraction, the architectural and scalable benefits of the framework persist regardless of the abstraction quality.

4.7.2 Privacy through Local Execution

Beyond data abstraction, the framework enforces physical data sovereignty. By locating the Simulation Module and Logic Unit on the edge node (e.g., the EV's onboard computer or a private home server), the raw data never crosses the network boundary. This eliminates the risk of "data leaks" at the central server level, which is a primary vulnerability in cloud-based Digital Twins where a single

breach can expose millions of users.

4.7.3 Comparison with Centralised Architectures

The privacy footprint of this framework offers a distinct advantage over the centralised baseline:

Feature	Centralised Cloud DT	Proposed Decentralised DT
Data Exposure	Raw Sensor Logs (GPS, Usage Patterns)	Aggregated Decisions (Power Profiles)
Data Storage	Third-party Cloud Database	Local Owner-Controlled Storage
Inference Risk	High (Direct access to user habits)	Low (Ambiguous intention signals)
Trust Requirement	High (Must trust the central admin)	Low (Trust in protocol only)

Table 4.1: Privacy comparison between Centralised and Decentralised Digital Twin frameworks.

4.8 Prototype Implementation

To validate the proposed framework and the subsequent algorithms, a comprehensive prototype was developed. The implementation focuses on creating a high-fidelity simulation environment that mirrors the decentralised nature of the architecture while enabling the training of the MARL agents.

4.8.1 Software Stack

The prototype is built primarily using **Python**, chosen for its extensive ecosystem of scientific computing and machine learning libraries. The software stack is organised into three layers:

- **Simulation Layer:** The physical dynamics of the EVs and the power grid are modeled using a custom Python-based discrete-event simulator. This simulator tracks the SoC, energy consumption, and grid load at 15-minute intervals.
- **Communication Layer:** The prototype leverages the **ZeroTier**[147] network virtualisation platform to create a secure, peer-to-peer virtual network. Each DT agent joins a common ZeroTier network, receiving a unique virtual IP address. This creates a flat, global Ethernet Local

Area Network (LAN), allowing agents to communicate directly as if they were on the same physical network and abstracting away the complexities of NAT traversal. Communication follows a broadcast pattern over User Datagram Protocol (UDP). When an agent needs to broadcast its Decision Information (I_{dec}), the data structure is serialised into a lightweight format like JSON and sent as a UDP broadcast packet to the virtual network. All other agents listen on a designated port, receive these packets, deserialise the information, and pass it to their Decision Maker for processing. This provides a realistic implementation of the P2P communication protocol.

- **Intelligence Layer:** The internal logic of the DT agents (the Decision Maker) is implemented using **PyTorch** [8]. This deep learning framework supports the construction and training of the neural networks required for the DT-MADDPG algorithm in Chapter 6. The interaction between the agents and the simulation follows the **OpenAI Gymnasium** [123] interface standard, ensuring modularity and compatibility with standard RL baselines.

4.8.2 Execution Environment

The simulation is designed to run in a distributed manner. While training is performed on a centralised high-performance computing cluster (utilizing GPU acceleration for the neural networks), the execution logic is decoupled. Each DT agent acts as an independent process instance, receiving only the local observation and the "broadcast" messages from its neighbors in the logical graph, strictly adhering to the privacy constraints defined in Section 4.2.

4.8.3 Validation Methodology, Assumptions, and Limitations

The prototype serves as the testbed for the experimental evaluations in subsequent chapters. The validation methodology focuses on quantifying performance under controlled conditions, which necessitates acknowledging the underlying assumptions and limitations of the implementation.

Performance and Scalability Validation

To empirically validate the framework, a series of controlled experiments are designed. The primary goal is to quantify the relationship between network size (N) and two key performance indicators:

- **Communication Overhead:** Measured by the average number of messages broadcast per agent per time-step. This tests the efficiency of the gossip protocol and localised communication, with the objective of demonstrating manageable, linear scaling.
- **Coordination Effectiveness:** A domain-specific metric measuring the quality of the collective outcome (e.g., grid load variance in the V2G case study). Stable performance as N grows indicates that the decision-level sharing remains effective at scale.
- **Computational Performance:** While communication is the primary bottleneck in distributed systems, we also consider the computational load (CPU and memory) on individual agents. The prototype allows for profiling the cost of local simulation and decision-making, confirming that the decentralised design avoids the processing bottlenecks of a central server.

Assumptions and Limitations

The prototype implementation, while comprehensive, operates under a set of assumptions that define the scope of this thesis and highlight areas for future work.

- **Idealised Network Conditions:** The prototype's communication layer, while using a realistic P2P virtualisation tool (ZeroTier), runs on a high-speed local network or a single machine. The experiments therefore do not account for real-world Wide Area Network (WAN) conditions such as packet loss, high latency, or network jitter. While the protocol is designed for resilience (e.g., considering the best-effort property of UDP and iterative state updates), its performance under these adverse conditions is not quantitatively evaluated in this study.
- **Simplified Adversarial Model:** The framework's robustness is tested against non-malicious

faults and simple forms of unreliability (e.g., a peer broadcasting consistently inaccurate predictions, as detailed in Chapter 5). The prototype does not, however, model sophisticated, strategic adversaries who might attempt to subtly manipulate the system for economic gain or to launch targeted attacks. Evaluating the trust mechanism's resilience against such advanced adversarial behaviours remains a key area for future work.

- **Model Fidelity Assumption:** The validation relies on the assumption that the simulation models (both the local agent models and the global environment model) are reasonably accurate representations of their physical counterparts. The "Sim-to-Real" gap, where unmodeled physics or environmental factors cause a divergence between simulation and reality, is a known challenge in all DT research. While the framework is designed to be robust to some level of noise, a full validation with Hardware-in-the-Loop (HIL) testing is beyond the scope of this thesis.

These limitations are explicitly acknowledged to properly contextualise the experimental results presented in the following chapters.

4.9 Summary

This chapter has presented the design of a fully decentralised DT framework, serving as the structural foundation for "Cooperation". By shifting from traditional data-level integration to a **Decision-Level Information Sharing** mechanism, the framework addresses the critical challenges of scalability and privacy in multi-owner Large Complex Systems.

We defined the network topology as a peer-to-peer graph and detailed the internal architecture of the DT node, which separates sensitive raw data from broadcast decision information. The analysis demonstrated that this abstraction effectively preserves user privacy while enabling the necessary information flow for system-wide visibility.

However, providing the "pipes" for information exchange is only the first step. In a decentralised

open system, agents must also account for uncertainty and the potential unreliability of their peers. Therefore, the next chapter introduces the **Trust Mechanism**, a critical component that ensures the "cooperation" remains robust and secure against error or malicious behavior.

Chapter 5

Digital Twin Collaboration on Decision Level

5.1 Introduction

The architectural framework presented in Chapter 4 established the structural foundation for a decentralised DT network. By organizing DTs in a peer-to-peer topology, we addressed the scalability limits of centralised control. However, connectivity alone does not constitute collaboration. For autonomous agents to operate as a cohesive LCS, they must actively exchange information to align their actions.

This introduces a critical duality of challenges. First, in a multi-stakeholder environment (such as a smart city or power grid), agents are often hesitant to share the sensitive raw data required for traditional centralised optimisation due to privacy and security concerns. Second, even if information is shared, the open nature of the network means peers may be unreliable. Inaccurate predictions originating from sensor faults, poor internal models, or malicious intent, can propagate through the network, destabilizing the system.

To address these challenges, this chapter introduces the "Decision Level Collaboration" strategy. We

propose a middleware-based framework that enables DTs to collaborate by exchanging high-level *predictions* and *decisions*, serving as the system's Minimum Viable Information (MVI) rather than raw data. To ensure robustness, this exchange is governed by a dynamic **Trust Mechanism** that continuously verifies peer reliability against physical reality.

The remainder of this chapter is organised as follows: Section 5.2 establishes the theoretical hierarchy of information sharing. Section 5.3 details the middleware architecture designed to facilitate this exchange. Section 5.4 defines the formal communication protocol. Section 5.5 presents the mathematical formulation of the Trust Mechanism. Finally, Section 5.6 validates the entire framework through a comprehensive VPP case study.

5.2 Theoretical Framework: From Data to Decisions

To enable collaboration without compromising the privacy of individual stakeholders, it is necessary to formalise the levels of information abstraction within the network. This thesis moves beyond the traditional "Data-Level" integration found in centralised systems to a "Decision-Level" paradigm.

5.2.1 Hierarchy of Information Sharing

We distinguish between three levels of interoperability in Digital Twin networks:

1. **Data Level:** The exchange of raw sensor readings and historical logs. While this allows for the highest fidelity of centralised modeling, it requires high bandwidth and exposes the internal state of the physical asset, maximizing privacy risk.
2. **Information Level:** The exchange of processed states or aggregated metrics. This offers reduced bandwidth but still often reveals the current status of the asset.
3. **Decision Level:** The exchange of *future intentions* (predictions) and *committed actions* (deci-

sions). This level creates a temporal decoupling: agents share what they *will do* or what they *expect to happen*, rather than the raw physics of *why* it is happening.

5.2.2 Internal vs. Interfacing States

A key theoretical contribution of this framework is the explicit classification of state variables to enforce the decision-level paradigm. Within any Digital Twin instance, states are categorised as follows:

- **Internal States:** These are attributes that are strictly private to the specific instance. They may be security-sensitive (e.g., a user's travel schedule or unique battery chemistry) or simply irrelevant to peers. These states reside solely within the local DT and are never transmitted.
- **Interfacing States:** These are states located at the boundary of the subsystem that are visible to and influential upon other DTs. An example is the electricity flow at the connection point between a house and the grid. While the internal reason for a power surge (e.g., "I am charging my car") is an Internal State, the surge itself ("I am drawing 10kW") is an Interfacing State.

Selection of Interfacing States

The selection of Interfacing States is not an automated process but a critical, domain-dependent design decision made during the establishment of the Digital Twin federation. The process is guided by a principle of "minimal necessary information," where the goal is to identify the smallest set of variables that are both necessary and sufficient for effective coordination.

The selection is typically based on a physical or logical system analysis to identify points of common coupling or interaction boundaries. For example:

- In an **electrical grid**, the Interfacing States are naturally the physical quantities at the point of common coupling where an asset connects to the grid, such as power flow (kW), voltage

(V), and frequency (Hz). These are the only variables through which one agent's actions can physically influence another.

- In a **logistics network**, an Interfacing State for a delivery vehicle might be its estimated arrival time at a shared distribution hub, as this directly affects the scheduling of other vehicles and warehouse resources.

Ultimately, the set of Interfacing States forms part of the standardised communication protocol that all participating agents must agree upon before joining the network. This pre-agreement ensures that all peers share a common understanding of the variables that govern their collective behaviour, enabling interoperability without exposing the private, internal logic that leads to changes in those states.

By restricting communication strictly to *Interfacing States*, specifically, predictions of their future values, the framework achieves two goals. First, it minimises the coupling between heterogeneous models; a peer needs to know *that* power is fluctuating, not *why*. Second, it preserves the "Willingness to Cooperate" by guaranteeing that sensitive Internal States remain fully under the owner's control.

5.3 Middleware Architecture for Collaboration

To facilitate decentralised collaboration without imposing significant restructuring on existing systems, the proposed framework is implemented as a generic distributed middleware. This middleware is a distinct framework component that operates alongside the Digital Twin in a separate process, rather than being deeply embedded within it. Each DT interacts locally with its own middleware instance via the DT's internal Interfacing Element (introduced in Chapter 4), while the middleware instances handle all the complex peer-to-peer interactions across the network using broadcasting and multicasting.

The internal architecture of the middleware is composed of five key components, as illustrated in the system design in Figure 5.1:

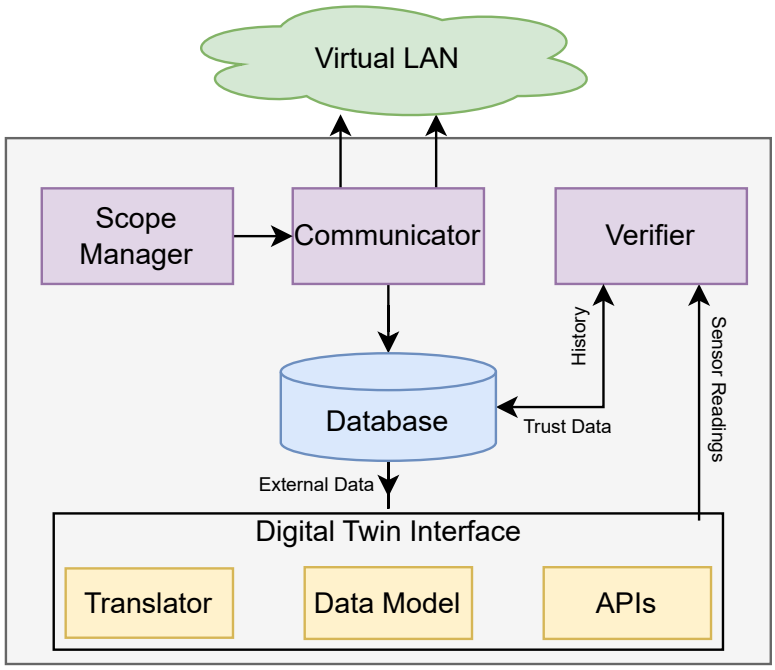

Figure 5.1: System Design of the Collaborative Middleware Layer.

5.3.1 The Communicator

The Communicator acts as the central conduit for information exchange within the collaborative network. It serves as a standardised interface, providing a unifying communication layer regardless of the specific domain or application context of the underlying DT. To ensure a secure and manageable infrastructure, the Communicator leverages virtualised network technology to place participating DTs within the same Virtual LAN, thereby abstracting the complexities of physical network discovery.

Conceptually, this Virtual LAN operates as an encrypted, peer-to-peer overlay network, effectively functioning as a decentralised smart Ethernet switch. Each DT agent joins the overlay and is assigned a unique, static virtual IP address from a private address range. This creates a flat, global logical network, allowing agents to communicate directly regardless of their underlying physical network topologies. To ensure a secure and resilient infrastructure, the overlay handles NAT traversal and firewalls conceptually through distributed hash tables or brokering nodes, while ensuring data traffic flows directly peer-to-peer to minimise latency and avoid central bottlenecks. All traffic across the

overlay is end-to-end encrypted, ensuring that communication between Digital Twins remains secure and private from external observers. This conceptual architecture provides the necessary P2P connectivity, robust routing, and strong encryption required for implementing the communication layer of the proposed decentralised framework.

5.3.2 The Digital Twin Interface (Translator)

A major challenge in LCSs is heterogeneity; different DTs (e.g., a wind turbine vs. a battery storage unit) utilise different internal data formats and protocols. The DT Interface addresses this by functioning as a *translator*. It adapts the domain-specific internal states of the DT into the framework's common standardised protocol (the Base Model discussed in Section 5.4), ensuring clear and consistent interpretation of messages across the federation.

5.3.3 The Scope Manager

In large-scale networks, broadcasting every message to every peer is inefficient and creates unnecessary overhead. The Scope Manager is responsible for optimising communication traffic by ensuring that messages are only sent to relevant peers. It achieves this by creating logical communication "scopes" based on the physical or logical relationships between agents.

Relevance is determined by analysing metadata associated with the Interfacing States. Each agent, upon joining the network, declares its context, for example, which electrical feeder it is connected to, which geographical zone it operates in, or which distribution hub it is heading towards. The Scope Manager uses this metadata to partition the network into dynamic, overlapping multicast groups.

For instance, in the VPP case study, an EV's prediction about its power demand is only physically relevant to other agents connected to the same local transformer, as their collective actions might overload it. Therefore, the Scope Manager would create a multicast group for all agents on "Feeder A" and another for "Feeder B". When an agent on Feeder A broadcasts a prediction, the message is

only sent to other members of the Feeder A group.

This content-based routing mechanism ensures that information is localised to the agents that are directly impacted by it, drastically reducing network traffic and improving scalability. It transforms the communication pattern from an inefficient global broadcast to a series of targeted, efficient multicasts.

Handling Peer Unreachability and Dynamic Topologies

The framework is designed for dynamic environments where peers may join, leave, or become temporarily unreachable. The middleware handles this through a combination of liveness checks and the inherent properties of the trust mechanism.

- **Liveness Detection:** The Communicator component implicitly monitors peer liveness. In a broadcast-based protocol, a peer is considered active as long as it periodically sends its prediction or decision messages. If a peer fails to broadcast any information within a predefined timeout period (e.g., several decision cycles), the Scope Manager flags it as inactive or unreachable.
- **Dynamic Scope Management:** Once a peer is flagged as unreachable, the Scope Manager dynamically removes it from all relevant multicast groups. Subsequent messages are no longer routed to the inactive peer, preventing wasted resources and timeouts. The system gracefully degrades, continuing to coordinate with the remaining active peers. When the peer comes back online and resumes broadcasting, it is automatically re-integrated into the appropriate scopes.
- **Trust-Based Resilience:** The Trust Mechanism (detailed in Section 5.5) provides a second layer of resilience. A peer that becomes unreachable stops providing predictions. The Verifier, unable to map these non-existent predictions to reality, will cause the peer's trust score to decay over time. This ensures that even if the network layer has a delay in detecting the failure, the decision-making layer will naturally start to ignore the unreliable peer, safeguarding the

system's integrity.

This two-pronged approach ensures that the collaborative network is robust against the common failures and dynamic changes inherent in large-scale, decentralised systems.

5.3.4 The History Database

Trust requires memory. The History Database stores the "collective memory" of the system, acting as a time-series repository that archives historical data on the predictions and decisions made by both the local DT and its peers. Crucially, it tracks the evolution of *Interfacing States* over time by continuously logging the incoming sensor telemetry from the physical environment alongside the previously broadcasted predictions from peers. Each entry is strictly indexed by its temporal metadata (timestamps), creating a synchronous mapping between what was predicted to happen at time t and what actually manifested in the physical world at time t. This structured temporal tracking is the foundational element for the trust mechanism, enabling the Verifier to perform retroactive analysis of peer reliability.

5.3.5 The Verifier

The Verifier is the enforcement engine of the trust mechanism. It is responsible for safeguarding system accuracy by meticulously comparing the historical predictions stored in the database against the subsequent real-world sensor data. This verification process identifies inconsistencies or potential faults, allowing the system to quantify the reliability of its collaborators.

Assumptions, Scope, and the Concept of Overlap

A fundamental assumption underlying this verification process is the existence of **overlap** between the verifying agent and the predicting peer. This overlap does not necessarily imply shared ownership of a physical asset, but rather *shared observability* of a common physical environment or logical

boundary. A local DT cannot verify a peer's prediction unless it possesses an independent means to measure or infer the predicted state once it manifests in the physical world.

Specifically, objective verification relies on the following foundational assumptions:

- **Shared Physical Observability:** The verifier must have access to independent sensor telemetry (H_R) that captures the physical domain where the peer's predicted Interfacing State occurs. For instance, in an electrical grid, if an EV agent predicts its future power injection, its peers can verify this claim if they are connected to, and can measure, the aggregated power flow at the same local transformer.
- **State Correlation and Inference:** The verifying agent does not necessarily need a sensor directly attached to the peer. Instead, its local sensors must capture data that is highly correlated with the peer's prediction. If a peer claims to inject 5kW, the verifier must be able to observe a corresponding 5kW shift in the shared environment's net state, assuming other background variables remain relatively constant or known.

Scope of Verification (Degrees of Overlap): The extent to which trust can be calculated, and the granularity of the resulting trust score, is scoped by the degree of this overlap:

- **Direct Overlap (High Verifiability):** Both agents measure the exact same physical coupling point (e.g., two energy assets on the same direct power line). Individual predictions can be verified with high precision, allowing for strict, individualised trust scores.
- **Aggregated Overlap (Partial Verifiability):** Agents share a broader environment (e.g., a community microgrid). A verifier might only see the *aggregate* effect of several peers. In this case, verification becomes statistical; if the aggregate physical reality matches the sum of aggregate predictions, the group's general trust is maintained. However, isolating a single minor faulty peer within the aggregate requires tracking patterns over a longer time horizon.
- **Zero Overlap (Unverifiable):** If agents operate in completely disjoint physical domains (e.g.,

EVs in different cities), they cannot physically verify each other's predictions. In such cases, direct objective trust calculation is impossible. The system must rely on topological isolation- where the Scope Manager naturally prevents disjoint agents from coordinatingensuring that unverifiable claims do not impact local decision-making.

Without this physical or logical overlap, predictions remain completely opaque, preventing the establishment of the objective, physics-grounded computational trust that this framework relies upon.

5.4 The Collaborative Protocol

Effective collaboration hinges on a shared language. The protocol defines both the structure of the information exchanged and the rules of interaction.

5.4.1 The Base Model

The foundation of communication is the *Base Model*, which defines how real-world entities are represented within the network. A "State" is defined as a tuple consisting of three elements: a globally unique identifier for clear identification, a human-readable name for interpretability, and the actual value reflecting the current status.

$$S = \langle ID, Name, Value \rangle$$

While states share logical or physical relationships (e.g., voltage and current on the same line), these relationships are modelled internally within each DT rather than in the protocol itself. We assume the list of states is agreed upon by all participating DTs prior to operation.

5.4.2 Information Structures

To practically implement the exchange of **Decision Information (I_{dec})** (representing the MVI) introduced in Chapter 4, the framework translates these abstract intentions into two primary structured

message types:

1. Prediction Tuples: A prediction represents a forecasted future value of a state. It is formatted as a tuple containing the prediction timestamp, the state identifier, and the predicted value:

$$P = \langle t_{pred}, ID_{state}, V_{pred} \rangle$$

To optimise bandwidth, predictions belonging to the same time frame or referring to the same state across different times are grouped into sets before transmission.

2. Decision Tuples: A decision is defined as an *action* coupled with its *related predictions*. An action represents a committed change to a specific state at a future time. Messages carrying decisions are identified with distinct headers but follow a similar structure to predictions. Crucially, a decision message bundles the action with the expected consequences (predictions), providing peers with the context needed to evaluate its impact.

5.4.3 Interaction Dynamics

The collaboration follows a continuous, cyclical exchange process:

1. **Broadcast Predictions:** Periodically, each DT broadcasts its current set of predictions to the federation.
2. **Receive and Update:** Participating DTs receive these predictions. Their internal Decision Makers utilise this external information (updated Interfacing States) to refine their own local models. This improves local accuracy and enables proactive planning based on high-confidence forecasts from peers.
3. **Broadcast Decisions:** Upon reaching a new local decision, the DT broadcasts both the decision itself and the associated updated predictions.

4. **Cycle:** Collaborators receive this decision, update their state, and the cycle continues.

Message Reliability and Protocol Choice

The protocol prioritises low-latency communication suitable for real-time systems over guaranteed message delivery. The prototype implementation utilises the UDP for broadcasting messages across the ZeroTier virtual network. UDP is a connectionless protocol that does not provide the delivery guarantees, ordering, or retransmission mechanisms of protocols like Transmission Control Protocol (TCP).

This choice is deliberate. In a high-frequency system with many agents, the overhead of establishing and maintaining numerous TCP connections would be prohibitive. Furthermore, the iterative nature of the protocol provides inherent resilience to occasional packet loss. Since agents continuously broadcast their updated state and predictions, a lost message is quickly rendered obsolete by a subsequent message in the next decision cycle.

Reliability is therefore handled at the application and trust layers, not the transport layer:

- **Idempotency:** The continuous broadcast of state means the system is self-correcting. The effect of a lost packet is transient.
- **Trust-Based Filtering:** As described in Section 5.5, a peer whose messages are consistently lost will fail to have its predictions verified against reality. This will cause its trust score to decay, naturally reducing its influence on the decisions of other agents.
- **Liveness Detection:** A peer that stops sending messages, whether due to failure or network partition, is eventually flagged as inactive by the Scope Manager (as detailed in Section 5.3.3) and temporarily removed from consideration.

This approach trades transport-layer reliability for lower latency and scalability, while building system-level robustness through higher-level mechanisms.

5.4.4 Locality and Scalability

To address scalability in large networks, the protocol leverages the concept of *locality*. Not every DT requires every message. By leveraging the mapping between global states and specific instances, the middleware identifies relevant collaborators for each specific prediction or decision. Consequently, broadcasts are targeted (multicasted) only to these relevant subsets rather than flooding the entire network, significantly reducing communication overhead.

5.5 The Trust and Verification Mechanism

While sharing predictions and decisions fosters knowledge exchange among collaborating DTs, it also introduces potential vulnerabilities. Even in trusted environments free from malicious actors, errors in predictions can occur due to model inaccuracies or sensor noise. To mitigate this risk, the framework incorporates a trust mechanism that assesses the accuracy of shared predictions and establishes a credit history for each DT instance. This information empowers the Decision Makers to weigh information from other instances more effectively.

5.5.1 Conceptualising Trust and Verifiability

To rigorously evaluate peer reliability, it is necessary to ground the mechanism in established definitions of trust. In sociological and cognitive schools of thought, trust is defined as a subjective belief, expectation, or willingness to accept vulnerability regarding the future behaviour of another entity under conditions of risk and interdependence [21]. In computational systems, this subjective expectation is operationalised as *computational trust*, a calculable metric derived mathematically from historical behaviour [54].

Crucially, this framework explicitly addresses the distinction between **trust**, **reputation**, and **verifiability**:

- **Verifiability (The Foundation):** Verifiability is a backward-looking, objective measurement. It is the definitive physical validation of whether a previously asserted prediction mathematically matched the subsequent physical reality measured by a sensor. It is a factual measurement of past events, completely free from subjective peer opinions.
- **Computational Trust / Reputation (The Outcome):** Trust is a forward-looking, probabilistic expectation. While traditional "reputation systems" often rely on subjective peer ratings (which are vulnerable to collusion), the computational trust score in this framework functions as an objective, *physics-grounded reputation*. It is the aggregated mathematical belief that future predictions will be reliable, derived strictly from past verifiability.

The Level of Trust: It is important to define the exact architectural level at which this trust operates. The trust evaluated here is strictly at the **Decision Level**. It does not evaluate "Data-Level" integrity (e.g., whether a raw sensor is perfectly calibrated) nor does it evaluate "Transport-Level" security (e.g., cryptographic signatures, which are handled independently by the overlay network). Instead, it evaluates the reliability of the *Interfacing States*the high-level predictions and intentions broadcasted by the peer's Decision Maker. By using objective verifiability as the engine to compute a dynamic computational trust score, the framework bridges the gap between past physical facts and future behavioural expectations at the decision layer.

5.5.2 Threat Model and the Role of Trust

In any open, decentralised system, information received from peers cannot be blindly trusted. The proposed trust mechanism is designed to provide resilience against a range of threats that fall into two broad categories:

- **Non-Malicious Faults:** These are unintentional errors that degrade the quality of information.
  - *Sensor Noise and Drift:* Physical sensors can provide inaccurate readings due to environ-

mental interference or degradation over time.

  - *Model Inaccuracy:* A DT's internal simulation model may be poorly calibrated or become outdated (model drift), leading to systematically flawed predictions.

  - *Transient Errors:* Sporadic software bugs or temporary network issues can cause a normally reliable peer to send incorrect data.

- **Malicious or Strategic Behaviour:** These are intentional actions by a self-interested or adversarial agent.

  - *Strategic Misinformation:* An agent might deliberately broadcast false predictions to manipulate the collective behavior for its own gain. For example, in a VPP, an agent could exaggerate its future energy demand to secure a larger share of available power at a lower price.

  - *Ignorance of Reality:* An agent might consistently provide predictions without regard for their accuracy, either due to negligence or a lack of incentive to maintain reliability.

  - *Free-Riding:* An agent might consume information from the network to improve its own performance without contributing reliable predictions of its own.

The trust mechanism addresses these threats not by cryptographic means (which verify identity), but by continuously assessing the *veracity* and *reliability* of the information content itself. It provides a "soft security" layer that quantifies the historical accuracy of each peer.

5.5.3 Verification Logic

The process of verifying predictions and calculating a trust score is handled by the dedicated *Verifier* component in the middleware. The procedure follows a four-step logic as outlined below:

1. **Prediction-Reality Mapping:** Periodically, the verifier reevaluates past predictions using newly

available sensor data. The algorithm iterates through all sensor readings (H_R) and previously recorded predictions (H_P). It compares the timestamps of these readings and predictions to identify those with a temporal difference that falls within a predefined threshold ($|t_{h_p} - t_{h_r}| \leq t_{thres}$). This threshold defines the maximum allowable time difference for a prediction to be considered relevant.

2. **Time Frame Aggregation:** To capture a snapshot of a DT's performance, the verifier groups all valid mappings into discrete time frames.

3. **Similarity Score Calculation:** For each time frame, the prediction accuracy is calculated using a similarity function. We utilise a Gaussian kernel function to normalise the Euclidean distance between the real measurement (h_r) and the prediction (h_p). The similarity score sim_i for a time frame F_i is given by Equation 5.1:

$$sim_i = \exp\left(-\frac{\sum_{(h_p,h_r)\in F_i}(h_p - h_r)^2}{2\sigma^2}\right) \tag{5.1}$$

where σ is the width of the Gaussian kernel, determined by the specific application scenario.

4. **Trust Score Calculation:** Leveraging the historical similarity scores, the system calculates a comprehensive Trust Score for each DT instance. This score incorporates both the accuracy (similarity) and the recency of the data. A time-decay weight w_i is applied to ensure that older similarity scores contribute less to the current trust value:

$$w_i = \frac{1}{1 + \alpha(t_{now} - t_i)} \tag{5.2}$$

The final Trust Score is the weighted average of the similarity scores over the past period:

$$Trust = \frac{\sum_{(t_i,sim_i)\in sim} sim_i \cdot w_i}{\sum w_i} \tag{5.3}$$

Generality and Typicality of the Algorithm:

It is important to note that while the mathematical primitives employed in this algorithm, specifically the Gaussian kernel for similarity and exponential time-decay weights, are typical and well-established in signal processing and conventional reputation systems [54], the overall verification pipeline represents a novel synthesis. Typical reputation models rely on binary, subjective peer feedback (e.g., successful or failed discrete transactions). In contrast, Algorithm 1 is designed as a *general* and objective verification engine. By abstracting the evaluation into configurable hyperparameters (time threshold t_{thres}, kernel width σ, and decay rate α), the algorithm is domain-agnostic. It can be applied to verify predictions of any continuous Interfacing Statewhether it be the power flow of an EV, the temperature of an industrial machine, or the speed of an autonomous vehicleby systematically mapping forward-looking digital predictions to backward-looking physical sensor telemetry.

5.5.4 Application of Trust Scores in Decision-Making

The calculated trust score is not merely a passive metric for observation; it is an active component in the decision-making loop of each agent. It allows an agent to intelligently filter and weigh incoming information from its peers, making the collective more resilient to the threats outlined above.

Trust scores are applied in two primary ways to influence an agent's decision-making process: through continuous, soft influence via weighting, and through discrete, hard influence via filtering.

Trust-Weighted Aggregation (Soft Influence)

The primary application is in the aggregation of peer predictions. When an agent needs to form a belief about the future state of the system (e.g., the total predicted load on a grid feeder), it computes a **trust-weighted average** of the predictions received from its neighbors. For a given future state, the agent's estimate V_{est} is calculated not as a simple average, but as:

$$V_{est} = \frac{\sum_{j \in \text{Peers}} \text{Trust}_j \cdot V_{pred,j}}{\sum_{j \in \text{Peers}} \text{Trust}_j} \tag{5.4}$$

where $V_{pred,j}$ is the value predicted by peer j, and Trust_j is that peer's current trust score.

The effect of this soft influence is twofold:

- **Influence Modulation:** Predictions from peers with a high trust score are given significant weight and strongly influence the agent's decisions. Conversely, predictions from peers with low trust scores (whether due to consistent inaccuracy or malicious behavior) are effectively marginalised. Their influence diminishes gracefully as their reliability decreases.
- **Graceful Degradation:** If a peer goes offline or starts sending nonsensical data, its trust score will rapidly decay. Its influence on the collective decision-making process diminishes automatically, without requiring explicit removal or administrative intervention. This makes the system robust and self-healing.

Threshold-Based Filtering (Hard Influence)

For applications with low tolerance for risk, a more aggressive "hard influence" strategy can be employed. This involves setting a minimum **trust threshold** ($Trust_{min}$). Any peer whose trust score falls below this threshold is considered completely unreliable and is temporarily filtered out from the decision-making process.

In this model, the set of peers considered for aggregation is dynamically defined as:

$$\text{TrustedPeers} = \{j \in \text{Peers} \mid \text{Trust}_j \geq Trust_{min}\}$$

The weighted average calculation is then performed only over this trusted subset.

This filtering mechanism acts as a definitive circuit-breaker, completely preventing a consistently

malicious or faulty agent from having any influence on the system. The choice of $Trust_{min}$ is a critical design parameter that balances sensitivity and stability. A high threshold provides strong security but may risk incorrectly excluding a temporarily faulty but otherwise valuable peer. A low threshold is more inclusive but offers less protection.

By combining both weighting and filtering, the framework provides a flexible, two-layer defense. Weighting handles the nuanced, gradual changes in reliability, while filtering provides a hard backstop against catastrophic failures or persistent malicious activity. This ensures that the system's collective intelligence is built upon a foundation of proven reliability.

5.6 Case Study: A Virtual Power Plant

To evaluate the effectiveness of the proposed framework, a Virtual Power Plant (VPP) federation was developed as a case study. A VPP functions as a distributed power plant by aggregating the capacities of diverse Distributed Energy Resources (DERs) to augment overall power generation and meet community demand .

5.6.1 Agent Modeling

The experimental setup models each DER as a DT within a collaborative federation. Four distinct DER types are incorporated:

- **Renewable Sources (Solar Panels & Wind Turbines):** The power output of these units is inherently variable, depending on weather conditions like wind speed and solar irradiance. They are equipped with battery units for temporary energy storage to enable short-term adjustments.
- **Hydropower Turbines:** Output is dictated by the water flow rate, with limited direct control.
- **Diesel Generators:** These offer fully controllable power output through adjustments to fuel

Algorithm 1 Verification and Trust score calculation.

Input: Target DT d, prediction history H_P, sensor history H_R, relevant states S_d.
Parameters: Time threshold t_{thres}, time frame count n, time span $[t_{start}, t_{end}]$.
Initialise: Prediction-reality mapping $M \leftarrow \varnothing$, aggregated frames $F \leftarrow \varnothing$, similarity scores $sim \leftarrow \varnothing$.

▷ Step 1: Prediction-Reality Mapping
for each state $s \in S_d$ **do**
    $M_s \leftarrow \varnothing$
    for each (h_r, h_p) in $H_R \times H_P$ **do**
        Extract timestamps t_{h_r} from h_r and t_{h_p} from h_p.
        if $|t_{h_p} - t_{h_r}| \leq t_{thres}$ **then**
            $M_s \leftarrow M_s \cup \{(t_{h_r}, h_p, h_r)\}$
            ▷ Map each sensor reading to the closest prediction
        end if
    end for
end for

▷ Step 2: Time Frame Aggregation
Split timespan $[t_{start}, t_{end}]$ into n discrete time frames $\mathscr{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$.
for $j = 1$ to n **do**
    $F_j \leftarrow \varnothing$ ▷ Initialise mapping for time frame τ_j
    for each state mapping M_s in M **do**
        Add all pairs (h_p, h_r) from M_s to F_j where the timestamp $t_{h_r} \in \tau_j$.
    end for
end for

▷ Step 3: Similarity Score Calculation
for $j = 1$ to n **do**
    $sim_j \leftarrow \text{calculateSimilarity}(F_j)$ ▷ Using Eq. 5.1
    $t_j \leftarrow \text{time\_since}(\tau_j)$ ▷ Time elapsed since frame τ_j
    $sim \leftarrow sim \cup \{(t_j, sim_j)\}$
end for

▷ Step 4: Trust Score Calculation
Sort sim by elapsed time t_j in descending order.
$Trust \leftarrow \text{calculateTrustScore}(sim)$ ▷ Using Eq. 5.3

input, serving as the primary balancing agents.

5.6.2 Decision Logic

For this chapter's evaluation, the Decision Maker within each DT employs a simple load-following policy. For diesel generators, the policy adjusts power output dynamically based on the difference between current demand and available supply. For other Distributed Energy Resource (DER), a policy dictates when to charge or discharge battery storage to compensate for output fluctuations.

5.6.3 Simulation Setup and Parameters

To validate the framework, a discrete-event simulation environment was developed in Python. The simulation models a local community's power grid, including fluctuating demand and the power generation from various DERs. The key parameters used for the experiments are summarised in Table 5.1. These values were chosen to represent a small but diverse local grid, providing a challenging testbed for coordination and trust evaluation.

Table 5.1: Simulation Parameters for the VPP Case Study.

Parameter	**Value**
DER Composition (Small-Scale Scenarios)	
Solar Panel DTs	2
Wind Turbine DTs	4
Hydropower DTs	1
Diesel Generator DTs	2
Community Demand Profile	
Base Load	500 kW (synthetic)
Ramp-Up Event	Step increase to 800 kW
Trust Mechanism Parameters	
Prediction Threshold (t_{thres})	10 seconds
Gaussian Kernel Width (σ)	0.5
Time-Decay Factor (α)	0.1

5.7 Experimental Evaluation

To comprehensively evaluate our framework's effectiveness, we establish three distinct small-scale scenarios and one large-scale scenario to assess scalability. The first scenario simulates a power ramp-up event, where the community's power demand abruptly increases at the simulation's outset. The VPP must then elevate its power output to meet the new demand. The second scenario explores the VPP's response to failures. Here, we simulate the concurrent malfunction of two wind turbine DERs due to strong winds. The VPP must compensate for this unexpected loss of generation capacity and maintain sufficient power output to meet demand. The third scenario investigates the VPP's ability to detect and address inaccurate predictions. We simulate a situation where one wind turbine DER generates erroneous power predictions. The remaining DERs must recognise this anomaly and adjust their output accordingly to ensure the community's power needs are met. In the first two scenarios, we compare the baseline, where DTs act purely based on local goals, and the collaborative case, where DTs exchange information using the framework. The third scenario compares the framework without the trust mechanism to the framework with the trust mechanism enabled. Finally, the scalability test investigates the framework's overhead as the number of DERs increases. This test aims to verify the framework's capability to handle growing system complexity.

For clarity in Figures 5.2, 5.3, and 5.4, the power output on the Y-axis is normalised. This value represents the ratio of the VPP's total power output to the current community power demand. A value of 1.0 indicates that the VPP is perfectly meeting the target demand, while values below 1.0 signify a power shortfall.

5.7.1 Methodology and Statistical Considerations

The experiments are run on a laptop with an AMD Ryzen 7 7840HS CPU and 32GB of RAM. The simulation environment, while deterministic in its core logic, incorporates sources of non-determinism to reflect real-world conditions. Specifically, the power output of renewable sources (wind and solar)

is based on stochastic models, and network communication can introduce minor, variable latencies. To account for this variability and ensure the robustness of our findings, each experimental scenario was executed 10 times with different random seeds.

The results presented in the subsequent figures (e.g., Figure 5.2) depict the **mean performance** across these 10 runs. The variance across runs was observed to be low, indicating stable and repeatable behavior from the coordination mechanism. For clarity in the line plots, only the mean is shown; however, the low variance confirms that the observed performance differences between approaches are statistically significant and not artifacts of random chance.

A sensitivity analysis was also considered. While a full parametric sweep of all simulation variables (e.g., trust parameters σ and α) was beyond the scope of this evaluation, the chosen parameters in Table 5.1 were selected based on empirical tuning to ensure stable and representative system behaviour. Furthermore, the scalability analysis presented in Section 5.7.5 effectively serves as a sensitivity analysis with respect to network topology and communication frequency, demonstrating how system performance and overhead respond to changes in the number of peers and the message interval.

Computational Overhead

The decentralised nature of the framework distributes the computational load, but it is important to consider the overhead on each individual agent. The primary sources of computational cost are:

- **CPU Usage:** The most intensive task is the trust verification process, which involves iterating through historical predictions and sensor data to calculate similarity scores. In our simulation, this process is executed periodically (e.g., every few minutes of simulated time) rather than continuously, which keeps the average CPU load manageable. The core simulation logic of the DT itself also contributes to the load, but this is inherent to any DT implementation.
- **Memory Usage:** The main consumer of memory is the *History Database*, which archives

all received predictions and local sensor readings over time. The memory footprint of this database grows linearly with the number of peers and the length of the historical window. To manage this in a long-running system, practical implementations would employ strategies such as data down-sampling for older records or maintaining a fixed-size sliding window for the trust calculation, thus bounding the memory usage.

While the overhead is non-trivial, it is a direct trade-off for gaining decentralised robustness and trust. The cost is borne by individual agents, avoiding the single point of failure and processing bottleneck of a centralised verification authority. The scalability analysis in Section 5.7.5 focuses on communication overhead, which is often the primary bottleneck in distributed systems, but the computational load per agent remains relatively constant as the network grows, assuming the number of direct peers is managed by the Scope Manager.

Experimental Limitations

It is important to acknowledge the limitations of the current experimental setup to contextualise the results:

- **Idealised Network Conditions:** While the protocol is designed to be resilient to network imperfections (as discussed in Section 5.4), the simulation environment runs on a single machine. Communication via the ZeroTier virtual network introduces only minimal latency and no significant packet loss. The performance under real-world WAN conditions, with higher latencies and packet loss rates, is not quantitatively evaluated in this study.
- **Simplified Adversarial Model:** The trust evaluation in Scenario 3 models a peer with consistently inaccurate predictions, representing a faulty or naive agent. It does not model a sophisticated, strategic adversary. A strategic agent might provide subtly biased or intermittently false information to manipulate the trust scores and exploit the system for economic gain. Evaluating the trust mechanism's resilience against such advanced adversarial behaviours remains an area

for future work.

These limitations define the boundaries of the current evaluation and highlight key areas for future investigation in more complex, real-world deployment scenarios.

5.7.2 Scenario 1: Ramp-Up Agility

Figure 5.2 shows the power ramp-up process for the baseline (Base) and collaborative (CDT) approaches. The CDT demonstrates a significantly higher ramp-up speed, reaching the target power output in 105 seconds, whereas the baseline takes 175 seconds. This 40% improvement in agility is primarily due to the communication with the diesel fuel generator. The shared predictions allow other DTs to anticipate the generator's ramp-up time and take a more aggressive approach, using their energy storage units to bridge the initial power gap more quickly.

5.7.3 Scenario 2: Failure Resistance

Figure 5.3 demonstrates how collaboration improves failure resistance. In this scenario, two wind turbines fail sequentially. The goal is to maintain the VPP's maximum potential power output above the demand target. While both approaches experience a shortfall, the CDT's performance is superior. By sharing failure predictions, the CDT allows other agents to proactively use their energy storage, resulting in a significantly higher minimum power output and a shorter period of insufficiency compared to the reactive baseline.

5.7.4 Scenario 3: Trust Robustness (Inaccurate Prediction)

Figure 5.4 shows the evaluation of the effectiveness of the trust mechanism. In this figure, the Collaborative DT used in the previous experiment acts as the baseline, and is compared with Trust-enabled Collaborative DT (T-CDT). To gain deeper insights into the experiment's results, we additionally plot the weight assigned to the faulty DT. This weight reflects the level of trust that other DTs placed in

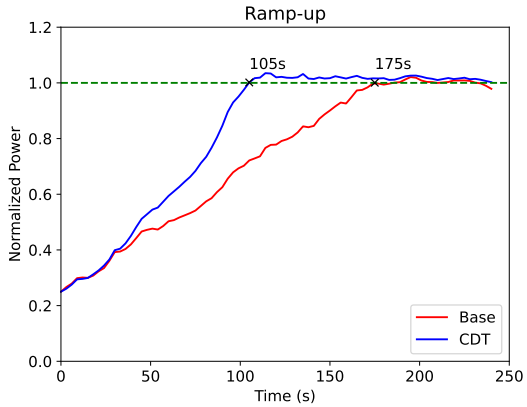

Figure 5.2: Results of the ramp-up scenario comparing the baseline and collaborative approaches, showing improved agility.

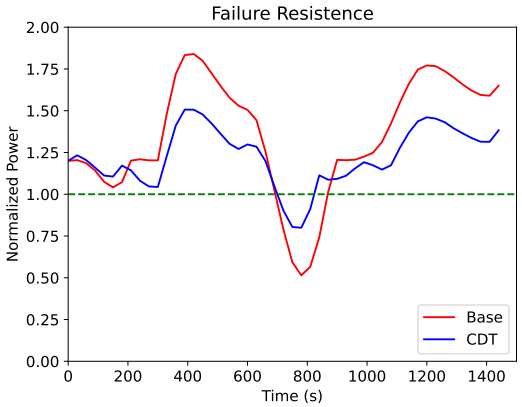

Figure 5.3: Results of the failure scenario comparing the baseline and collaborative approaches, showing improved failure resistance.

the faulty one over time. In the scenario setup, three periods of power insufficiency were planned. Firstly, it could be observed that due to the initial power insufficiency, the faulty DT could not provide the power output that it had predicted. Thus, the weight is rapidly reduced in all other instances, causing the decision makers in each DT to leave out more safety margins. When overall power output exceeds demand, the weight assigned to the faulty DT increases. This occurs because the faulty behavior is no longer readily apparent under these conditions. By incorporating a trust mechanism that adjusts the weight of contributions from individual DTs, we effectively mitigated the second period of insufficiency and prevented a third one, as observed in the figure.

5.7.5 Scalability Analysis

To evaluate the scalability of the framework, a series of experiments were conducted to analyse how system performance and communication overhead are affected by network size, connectivity density, and communication frequency. The key metrics are the communication overhead (messages per agent per time-step) and system performance (normalised power variance, where lower is better).

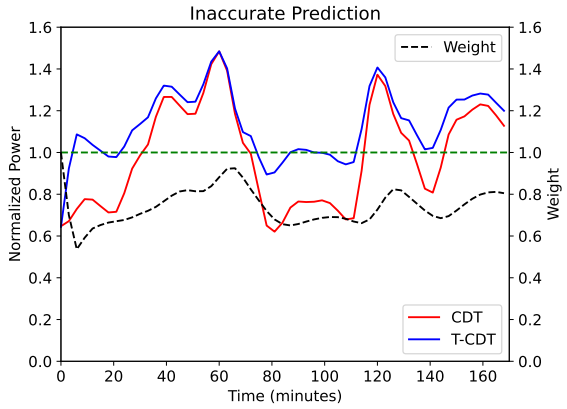

Figure 5.4: Results of the inaccurate prediction scenario comparing the plain collaborative approach and trust-enabled collaborative approach, showing improved robustness.

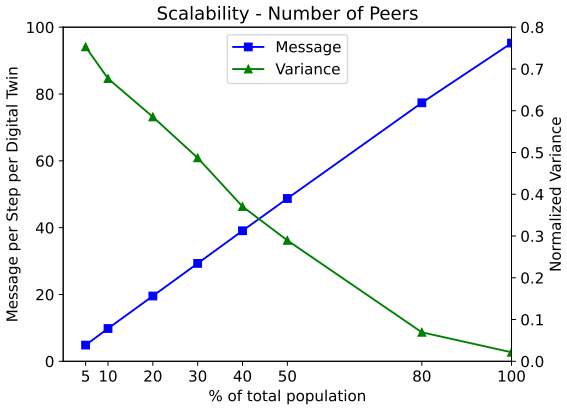

Figure 5.5: Results of the scenario varying the number of peers, showing the trade-off between communication overhead and system performance.

Impact of Peer Connectivity

Figure 5.5 illustrates the trade-off between performance and overhead as a function of peer connectivity. In this experiment, the total network size was fixed at 200 agents, and the number of peers each agent communicates with was varied from 5% (10 peers) to 100% (all 200 peers).

As expected, the communication overhead per agent increases linearly with the number of peers. More importantly, system performance (lower variance) improves significantly as agents gain visibility into a larger portion of the network. However, this improvement exhibits diminishing returns. The most substantial performance gain is achieved when moving from a small peer set (5-10%) to a moderate one (20-30%). Beyond this point, the marginal benefit of adding more peers decreases, while the communication cost continues to rise linearly. This finding is critical as it validates the architectural role of the *Scope Manager* (Section 5.3.3). It confirms that a "sweet spot" exists where agents can achieve near-optimal coordination by communicating with a limited, relevant subset of peers, avoiding the prohibitive cost of an all-to-all communication topology.

Impact of Network Size and Communication Frequency

Figures 5.6 and 5.7 analyse the impact of the total network size (N) and the communication frequency (controlled by the *event interval*). In this experiment, the number of peers for each agent was capped at $\max(10, 0.1 \times N)$.

Figure 5.6 shows that the communication overhead per agent is decoupled from the total network size. The overhead remains constant as long as the local peer count is capped (from $N = 10$ to $N = 100$, the peer count is 10). The overhead only begins to increase linearly when the network grows large enough ($N > 100$) for the 10% rule to exceed the fixed cap of 10. This demonstrates that the communication load on any individual agent is a function of its local neighborhood size, not the global population size, which is a key property of a scalable distributed system. As expected, the event interval has a predictable inverse relationship with overhead; more frequent communication (smaller interval) leads to more messages.

Figure 5.7 reinforces this finding from a performance perspective. System performance is primarily dictated by the communication frequency (event interval) and is largely insensitive to the total number of agents, provided the local connectivity is maintained. The initial performance improvement from $N = 5$ to $N = 10$ occurs because the system gains a sufficient number of agents to effectively manage the load. After this point, performance remains stable as the network grows, because each agent's decision-making context (its local neighborhood) remains consistent in size. The slight degradation in performance at very large network sizes ($N > 100$) can be attributed to the increased aggregate complexity and stochasticity of a larger system, but the overall stability is maintained.

Synthesis and Design Implications

The scalability analysis yields two key conclusions. First, the framework's performance is governed by local factors such as peer connectivity and communication frequency rather than the global network size. This confirms that the decentralised, multicast-based communication strategy successfully

avoids the bottlenecks of centralised systems. Second, there is a clear and tunable trade-off between performance and communication overhead. For a system operator, this provides crucial flexibility:

- In performance-critical applications, the peer count and communication frequency can be increased to maximise coordination, at the cost of higher network traffic.
- In resource-constrained environments (e.g., IoT devices with limited bandwidth), the peer count can be limited and the event interval increased, providing "good enough" coordination with minimal overhead.

Ultimately, these results demonstrate that the proposed framework provides a robust and scalable foundation for collaboration. The linear scaling of overhead within local groups, combined with the constant load per agent regardless of global size, confirms the architecture's suitability for deployment in large-scale, real-world systems.

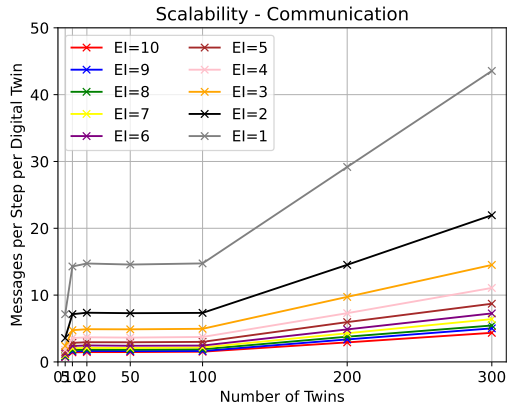

Figure 5.6: Communication Overhead vs. Number of Digital Twins, compared across different event intervals, showing scalability.

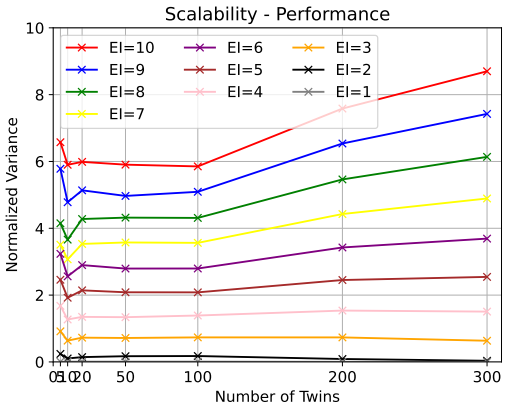

Figure 5.7: Performance vs. Number of Digital Twins, showing stability across different network sizes and event intervals.

5.7.6 Sensitivity Analysis of Trust Parameters

Figure 5.8 presents the sensitivity analysis of the trust mechanism's hyperparameters: σ (kernel width) and α (time decay). The performance metric used is *Normalized System Variance*, consistent

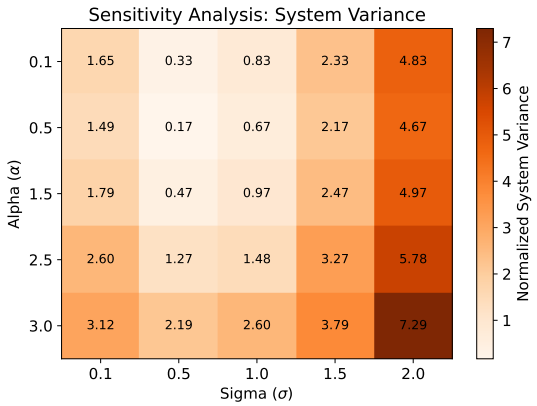

Figure 5.8: Heatmap showing the sensitivity of system variance to trust mechanism parameters σ and α. The optimal region (low variance) is centered around $\sigma = 0.5$ and $\alpha = 0.5$.

with the scalability experiments in the original study. A lower variance indicates a more stable Virtual Power Plant (VPP) that effectively manages power demand despite failures or inaccurate predictions.

Identification of the Parameter Sweet Spot

The results reveal a distinct region of optimal stability centered around $\sigma = 0.5$ and $\alpha = 0.5$, where the system variance is minimized to **0.17**. This confirms that the trust mechanism requires a balanced configuration:

- The parameter $\sigma = 0.5$ provides a sufficiently strict boundary to distinguish genuine faults from sensor noise (see Equation 1).
- Simultaneously, $\alpha = 0.5$ ensures that the system retains a “memory” of past failures long enough to prevent premature re-trusting of faulty nodes (see Equation 2).

Impact of σ (Sensitivity)

Moving right along the X-axis (increasing σ) results in a sharp degradation of performance.

- **High Sigma (The "Oblivious" Zone):** At $\sigma = 2.0$, the variance spikes significantly, reaching **7.29** when $\alpha = 3.0$. Mathematically, a large denominator in the Gaussian kernel causes the

similarity score sim_i to remain high even when the difference between prediction (h_p) and reality (h_r) is large. This renders the system blind to the wind turbine's inaccurate predictions, preventing the backup batteries from engaging effectively.

- **Low Sigma (The "Nervous" Zone):** Conversely, at very low sigma ($\sigma = 0.1$), the variance is moderately high (**1.65** at $\alpha = 0.1$). This indicates over-sensitivity, where minor sensor noise is penalized as a fault, causing unnecessary fluctuations in the VPP's decision-making.

Impact of α (Memory/Forgiveness)

Moving down the Y-axis (increasing α) generally increases system variance, particularly when combined with suboptimal σ values.

High Alpha (The "Amnesic" Zone): High α values (e.g., $\alpha = 3.0$) lead to poorer performance. According to Equation 2, a high α rapidly reduces the weight of historical data. This causes the system to "forgive" a faulty Digital Twin almost immediately after a momentary return to normal behavior. In the failure scenario, this likely results in oscillation, where the system repeatedly trusts and then distrusts the failing turbine, destabilizing the aggregate power output.

Summary

The heatmap demonstrates that the proposed framework is robust but benefits significantly from parameter tuning. The system performs best when configured to be moderately strict ($\sigma \approx 0.5$) with a balanced memory span ($\alpha \approx 0.5$). Extreme configurations either too lenient (High σ) or too forgetful (High α) undermine the benefits of the trust mechanism.

5.8 Discussion

The evaluation results highlight two critical findings. First, the "Decision-Level" collaboration protocol successfully decouples the need for raw data sharing from the ability to coordinate. By sharing only high-level intentions, the system achieved a 40% improvement in ramp-up agility without compromising privacy. Second, the Trust Mechanism proved essential for robustness. In open environments, "blind trust" is a vulnerability; the proposed similarity-based verification provides a mathematical shield against unreliability.

The evaluation results show that the proposed framework successfully boosted the system performance over the baseline case on two common scenarios, and the trust mechanism improved the system's robustness over bad predictions. The evaluation of the scalability shows the framework's potential to scale up while maintaining the performance and keeping the communication overhead in a reasonable range.

However, a limitation remains: the decision logic used in these agents was reactive and rule-based (load-following). While they shared information, they did not learn to optimise their policies over time. They simply reacted to the broadcasted intentions of others. To achieve true system-wide optimisation, the agents require intelligence - the ability to learn from interaction and maximise long-term rewards.

5.9 Summary

This chapter introduced a framework for "Decision Level Collaboration" in decentralised Digital Twin networks. We detailed the middleware architecture, the formal collaboration protocol, and a mathematical Trust Mechanism for verifying peer reliability. Through a Virtual Power Plant case study, we demonstrated that this approach significantly enhances system agility and fault tolerance while scaling efficiently.

Having established the *means* to communicate (Chapter 4) and the *safeguards* for reliability (Chapter 5), the final component of this thesis is the *intelligence* to act optimally. The next chapter introduces the Intelligent Digital Twin Collaboration framework, where we replace static rules with MARL to enable fully autonomous, optimised coordination.

Chapter 6

Intelligent Coordination via Multi-Agent Reinforcement Learning

6.1 Introduction

The previous chapters established the structural foundation for decentralised Digital Twin Networks (DTNs). Chapter 4 introduced the framework for privacy-preserving information exchange, and Chapter 5 defined the trust mechanism to ensure the reliability of that exchange. However, a robust communication channel and a trusted network are not sufficient to achieve optimal system performance. In the VPP case study presented in Chapter 5, agents utilised simple rule-based logic (e.g., load-following) to react to peer information. While effective for basic stability, these reactive strategies fail to capture the complex, long-term strategic consequences of actions in a dynamic environment.

To achieve true system-wide optimisation, such as maximising renewable energy utilisation while minimising costs, agents must possess intelligence. They must be able to learn emergent coordination strategies that balance their local self-interest with global system goals. MARL offers a powerful

paradigm for this "Intelligent Coordination".

However, applying state-of-the-art MARL algorithms like MADDPG to decentralised Digital Twin networks presents a fundamental conflict. Standard MADDPG relies on the CTDE paradigm, which typically requires a centralised critic to access the *raw* observations and actions of all agents. This requirement directly violates the privacy-preserving design constraints established in this thesis, where sensitive user data (e.g., EV travel schedules) must remain local.

To bridge this gap, this chapter introduces **Digital Twin Assisted Multi-Agent Deep Deterministic Policy Gradient (DT-MADDPG)**, an algorithm designed to resolve the conflict between privacy and intelligent coordination within the proposed framework. The core algorithmic contribution is a **simulation-assisted critic**, a hybrid mechanism that enhances the standard MARL training process. Instead of requiring a centralised critic to access raw private data, this approach leverages the *Collaborative Global Model* introduced previously, a high-fidelity simulation environment constructed from the privacy-preserving, decision-level information shared by the DTs. This allows the critic to gain the global visibility needed for stable training without violating data sovereignty. By integrating the predictive power of Digital Twins directly into the learning loop, DT-MADDPG enables agents to learn globally optimal policies in a fully decentralised and private manner, completing the thesis's proposed framework for intelligent coordination.

6.2 V2G Problem Formulation

Building upon the generic Decentralised Partially Observable Markov Decision Process (Dec-POMDP) framework established in Section 4.5 of Chapter 4, this section instantiates the mathematical model specifically for the V2G coordination problem and defines the corresponding reinforcement learning objectives.

6.2.1 State and Action Spaces

In the V2G context, the set of agents N represents the autonomous EV Digital Twins. The action space for each agent, $a_i \in A_i$, is a continuous value representing the flow of power at the current time. It is constrained by the maximum charging and discharging rates: $a_i \in [-a_{max,discharge}, a_{max,charge}]$. A negative value corresponds to discharging electricity from the EV into the grid, while a positive value represents charging the EV's battery.

As defined by the framework's generic observation function, an agent bases its decisions on a local observation $o_i \in \Omega_i$. In this V2G instantiation, this is a concatenated vector composed of:

- **Private Local State:** The agent's own internal variables, specifically its current battery level (SoC_i), the time remaining before the owner's scheduled departure (T_{ETD}), the owner's target SoC at departure (SoC_{tgt}), and the real-time energy price (P_t).
- **Aggregated Peer Information:** A "Context Vector" representing the collective state of the local grid, constructed by aggregating the MVI (predicted power profiles) shared by neighbors, weighted by the trust mechanism from Chapter 5.

The objective of each agent i is to learn a local deterministic policy $\mu_i : \Omega_i \rightarrow A_i$ such that the joint policy $\mu = \{\mu_1, \ldots, \mu_n\}$ maximises the expected cumulative discounted reward:

$$J(\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(t)\right] \tag{6.1}$$

The core components of our learning framework are defined by a set of functions. Each agent's behaviour is governed by a deterministic policy $a_i = \mu_{\theta_i}(o_i)$, represented by a neural network with parameters θ_i. This function maps the agent's local observation to a specific charge or discharge action. The action-value function is the centralised critic Q_i with parameters ϕ_i. It approximates the expected outcome after taking a joint action $a = \{a_1, ..., a_N\}$ in the global state s. The key distinction of the

proposed architecture is that the critic is enhanced by the prediction generated by the collaborative global model, denoted as s_{sim}. Therefore, the critic will be formulated as $Q_i = (s, a, s_{sim}) \to \mathbb{R}$. The collaborative global model function f_{sim} represents the global simulation model of the V2G network. It maps the current global state and a joint action to a predicted next state $f_{sim}(s, a) \to s_{sim}$.

6.2.2 Hierarchical Reward Structure

To complete the Dec-POMDP formulation, we must define the reward function $\mathscr{R}$. To address the complex trade-offs in the V2G network, we designed a hierarchical reward structure that balances system-level goals with the individual preferences of users. The final reward signal for an individual agent i at each time step t, denoted as $r_i(t)$, is a weighted combination of a shared global reward and the agent's own local reward. This is formally expressed as:

$$r_i(t) = w_{global} \times R_{global}(t) + (1 - w_{global}) \times R_{local,i}(t) \tag{6.2}$$

In this formulation, w_{global} is a hyper-parameter that controls the agent's alignment with the collective goal. The global reward component, $R_{global}(t)$, is shared among all agents and is derived from system-wide performance metrics. Specifically, it is designed to reward grid stability (minimizing the variance of power drawn from non-renewable sources) and maximizing the use of renewable energy. It is calculated as:

$$R_{global}(t) = -\alpha \cdot \mathrm{Var}(P_{grid}(t)) + \beta \cdot \frac{P_{used\_renewable}(t)}{P_{available\_renewable}(t)} \tag{6.3}$$

Here, $\mathrm{Var}(P_{grid}(t))$ is the variance of non-renewable power drawn from the main grid, and the second term is the utilisation ratio of available renewable power. The coefficients α and β are crucial scaling factors that serve two purposes. First, they normalise the two reward components, which have vastly

different numerical ranges. The power variance can be a large number (e.g., in the thousands), while the utilisation ratio is bounded between 0 and 1. Without scaling, the variance term would dominate the reward signal, preventing the agent from learning to optimise for renewable usage. Second, they encode the relative importance of the two objectives.

These coefficients are hyperparameters and are not learned by the agent. Their values are typically determined empirically through a combination of domain knowledge and hyperparameter tuning. For instance, an initial value for α can be estimated to bring the expected variance term to a scale comparable to the utilisation ratio (e.g., $\alpha \approx 1/\mathbb{E}[\text{Var}(P_{grid})]$). Subsequently, a grid search can be performed to fine-tune the relative weights of α and β to achieve the desired balance between grid stability and renewable energy maximisation. For the experiments in this chapter, these values were tuned to give roughly equal importance to both objectives during training. The local reward component, $R_{local,i}(t)$, represents the individual objectives of agent i.

Individual users typically pursue two distinct and competing goals: satisfaction of their State of Charge (SoC) needs and financial revenue. To model this, the local reward is a weighted sum of two terms. We use a single weight, w_{SoC}, to model a user's preference, with the constraint that the weights sum to one. The local reward for agent i is shown in Equation 6.4:

$$R_{local,i}(t) = w_{SoC} \times R_{SoC}(\text{terminal}) + (1 - w_{SoC}) \times R_{revenue}(t) \tag{6.4}$$

The first term, R_{SoC}, is a large terminal reward applied only at the agent's scheduled departure time. It is a large positive constant, $C_{success}$, if the final SoC has reached the user's desired threshold, and a large negative penalty, $-C_{fail}$, otherwise. The second term, $R_{revenue}(t)$, is the immediate financial outcome, calculated as:

$$R_{revenue}(t) = \text{Price}(t) \times P_{flow}(t) \tag{6.5}$$

Price(t) is the real-time electricity price, and $P_{flow}(t)$ is the power flow of the EV, which is positive for discharging (selling) and negative for charging (buying). This two-level reward structure provides a flexible and powerful mechanism for guiding the agents toward sophisticated, cooperative behaviours that respect both global system needs and individual user priorities.

6.3 System Architecture for Intelligent Collaboration

To enable intelligent coordination while respecting privacy constraints, we propose a hybrid architecture that integrates the Collaborative Digital Twin network with the CTDE paradigm. As illustrated in the system design, the architecture is composed of two primary layers: a decentralised execution layer of autonomous actors (the EV Digital Twins) and a centralised training layer.

6.3.1 The Decentralised Actor (EV Digital Twin)

Each Electric Vehicle in the network is represented by a local Digital Twin, which functions as an independent reinforcement learning actor. The internal structure of this actor includes :

- **Local Database:** Stores private state information such as current battery level (SoC), estimated mileage, and user preferences.
- **Decision Maker (Actor Network μ_i):** A deep neural network parametrised by θ_i that takes the local observation o_i as input and outputs a deterministic action (power flow).
- **Prediction Engine (Internal Simulator):** A physics-based model used for predicting future states locally.

Crucially, these actors operate in a fully decentralised manner during execution. They make decisions based solely on their local observations and the high-level decision information received from peers via the Communicator .

Table 6.1: Table of Notations for DT-MADDPG Algorithm

Notation	Description
N	Total number of agents (EVs) in the system.
i	Index for an individual agent, where $i \in \{1, \ldots, N\}$.
s	Global state of the environment, composed of all local observations.
o_i	Local observation for agent i.
a	Joint action, composed of all individual agent actions $\{a_1, \ldots, a_N\}$.
a_i	Action taken by agent i.
r_i	Reward received by agent i.
$\mathscr{N}$	Random noise to encourage exploration.
$\mathscr{D}$	Central replay buffer, which stores experience tuples.
$\mu_i(\theta_i)$	Actor network for agent i, parametrised by θ_i.
$Q_i(\phi_i)$	Complete action-value function for agent i.
$Q_{res,i}(\phi_i)$	Residual critic network for agent i, parametrised by ϕ_i.
f_{sim}	Collaborative Global Model, which functions as a system simulator.
R_{sim}	Value baseline calculated from the simulation rollout.
y_i	Target value used to train the critic for agent i.
μ'_i, $Q'_{res,i}$	Target networks for the actor and residual critic, respectively.
$R_{global}(t)$	Shared global reward component at time t.
$R_{local,i}(t)$	Local reward component for agent i at time t.
$R_{revenue}(t)$	Immediate economic reward component at time t.
$R_{SoC}(\text{terminal})$	Terminal reward for SoC satisfaction at the end of a session.
$P_{grid}(t)$	Power drawn from the non-renewable grid source at time t.
$P_{flow}(t)$	Power flow of an EV at time t (positive for discharging, negative for charging).
$\text{Price}(t)$	Real-time price of electricity at time t.
$C_{success}, C_{fail}$	Positive constants for the terminal SoC success reward and failure penalty.
w_{global}	Weight balancing the global and local components of the reward function.
w_{soc}	Weight balancing SoC satisfaction and revenue in the local reward function.
γ	Discount factor for future rewards.
τ	Soft update parameter for updating the target networks.
k'	Simulation rollout horizon (number of steps).

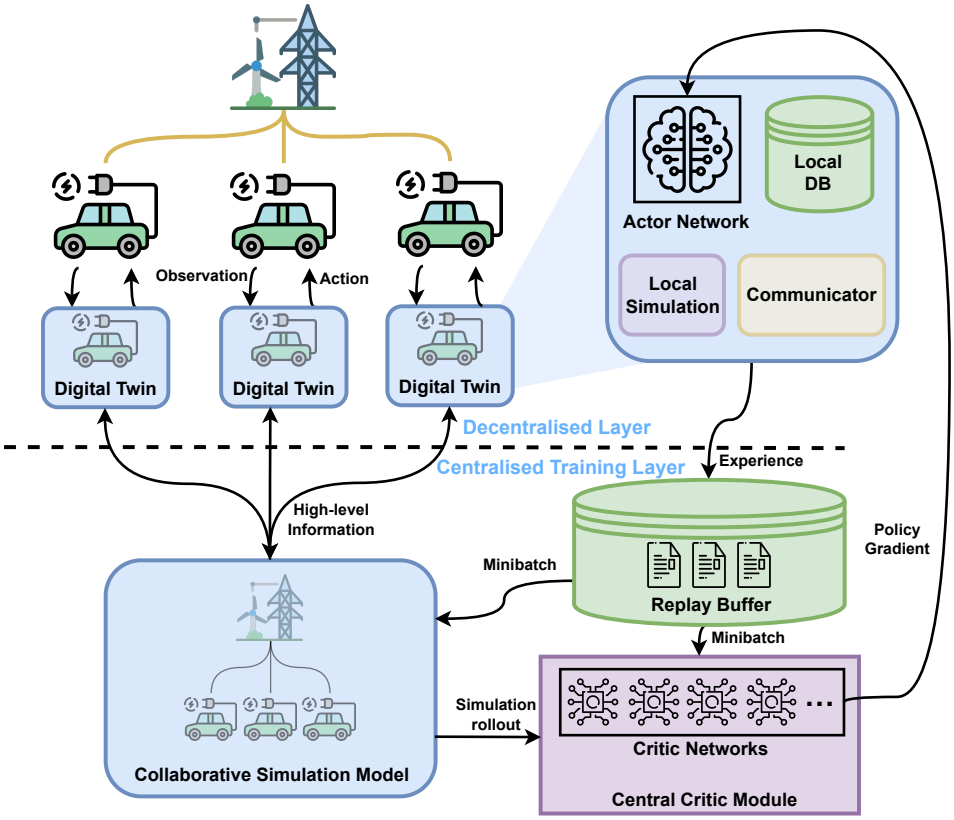

Figure 6.1: System Architecture of the DT-MADDPG Algorithm, showing the Decentralised Actor Layer and the Centralised Training Layer with the Collaborative Global Model.

6.3.2 The Collaborative Global Model

The bridge between privacy and centralised intelligence is the **Collaborative Global Model** (f_{sim}). This is a high-fidelity, physics-based simulator of the entire multi-agent environment (e.g., the V2G network) that is used exclusively during the centralised training phase.

Construction

The term "collaborative" refers to how the model is assembled. It is not learned from data, but rather constructed by aggregating the individual, physics-based models of each component in the system. It comprises two main parts:

- **Shared Environment Model:** A model of the common infrastructure, such as the power grid's physics, base load, and renewable generation dynamics. This part is typically static and defined by the system operator.
- **Agent Models:** A collection of the internal simulation models for each participating agent (e.g., the battery dynamics and efficiency model for each EV). In a real-world deployment, each Digital Twin would register a standardised version of its local model with the training service upon joining. For the purposes of this research, we assume these individual models are known and provided to the central trainer.

This construction process creates a complete, high-fidelity "proxy" of the real world that can be used for simulation, without requiring access to the agents' private, real-time state data.

Update Mechanism (During Training)

The model's parameters are static; it is not "updated" in the sense of being retrained. Instead, its *state* is updated iteratively during the "Simulation Rollout" phase of the DT-MADDPG algorithm (detailed in Section 6.4). When the critic needs to evaluate a state-action pair (s, a), the Collaborative

Global Model takes the current global state s and the joint action a as input. It then applies its internal physics equations to deterministically compute the next system state s'. This process is repeated for a short horizon to generate a predicted trajectory, which provides the critic with the necessary global information to guide the learning process while respecting the privacy boundaries of the individual agents.

6.4 The DT-MADDPG Algorithm

The core algorithmic contribution of this thesis is the **Digital Twin Assisted Multi-Agent Deep Deterministic Policy Gradient (DT-MADDPG)** algorithm. It introduces a novel "Simulation-Assisted Critic" that leverages the predictive capabilities of the Collaborative Global Model to enhance the learning process .

The fundamental premise of our approach is that physics-based simulation models are highly effective at making accurate predictions over short-term horizons, but their accuracy degrades over time due to compounding uncertainties. Conversely, reinforcement learning critics excel at learning long-term strategic values but suffer from sample inefficiency when learning physics from scratch. DT-MADDPG combines these strengths through Simulation Rollout and Value Decomposition.

6.4.1 Simulation Rollout

Instead of relying on a simple one-step prediction, the algorithm employs a multi-step lookahead mechanism. For a given joint state-action pair (s,a) sampled from the replay buffer, the Collaborative Global Model f_{sim} performs a Simulation Rollout for a horizon of k' time steps .

This process generates a predicted trajectory of future states and rewards: $(s_{t+1}, r_{t+1}, \ldots, s_{t+k'}, r_{t+k'})$. From this trajectory, we calculate the Simulation Baseline Value (R_{sim}), which represents the dis-

counted sum of rewards over the predictable short-term horizon :

$$R_{sim}(s,a) = \sum_{j=1}^{k'} \gamma^{j-1} r_{t+j} \tag{6.6}$$

where γ is the discount factor. This term (R_{sim}) effectively captures the "known physics" of the immediate future, providing a stable ground-truth anchor for the critic.

6.4.2 Value Decomposition

To account for the strategic value beyond the simulation horizon, we reframe the learning task using Value Decomposition. We decompose the total action-value function $Q_i(s,a)$ into two distinct components :

1. **The Physics-Based Component (R_{sim}):** The explicit short-term value calculated by the simulation rollout (as defined above).
2. **The Residual Component ($Q_{res,i}$):** A learned value that captures the long-term strategic return beyond the simulation horizon k'. This component is approximated by a deep neural network (the Residual Critic) parametrised by ϕ_i.

The complete Q-value function is thus expressed as the sum:

$$Q_i(s,a) = R_{sim}(s,a) + Q_{res,i}(s,a) \tag{6.7}$$

By explicitly providing the physics-based value R_{sim}, the neural network $Q_{res,i}$ is relieved of the burden of learning basic system dynamics. Instead, it focuses solely on learning the "residual" error between the simulation and the true infinite-horizon value, leading to faster convergence and greater stability .

6.4.3 Training Procedure

The training process integrates these concepts into the standard actor-critic loop, as detailed in Algorithm 1. The centralised training phase proceeds as follows:

1. **Sampling:** A random minibatch of experiences is sampled from the central Replay Buffer.
2. **Rollout Calculation:** For each sample, the Collaborative Global Model f_{sim} executes the simulation rollout to compute the baseline R_{sim}.
3. **Target Calculation:** The target value y_i for the critic update is computed using the target networks $(\mu_i', Q'_{res,i})$ to ensure stability:

$$y_i = r_i + \gamma(R_{sim,next} + Q'_{res,i}(s', a')) \tag{6.8}$$

   where $R_{sim,next}$ is the rollout value from the next state s'.
4. **Critic Update:** The Residual Critic network $Q_{res,i}$ is updated by minimizing the mean squared error loss between the target y_i and the composite Q-value estimate :

$$\mathscr{L}(\phi_i) = \mathbb{E}\left[(y_i - (R_{sim}(s,a) + Q_{res,i}(s,a)))^2\right] \tag{6.9}$$

5. **Actor Update:** The actor policy μ_i is updated via the deterministic policy gradient, utilizing the gradient of the full composite Q-function with respect to the action.

6.4.4 Detailed Algorithmic Walkthrough

The DT-MADDPG algorithm, outlined in Algorithm 2, integrates physics-based simulation into the MARL training loop. The process can be broken down into the following key stages:

1. Initialisation: The process begins by initializing the necessary components for each of the N

Algorithm 2 Digital Twin-Assisted MADDPG (DT-MADDPG)

Input: Number of agents N, simulation rollout horizon k', hyper-parameters γ, τ.
Parameters: Number of training episodes M, max steps per episode, minibatch size K, exploration noise process $\mathcal{N}_t$.
Output: Trained decentralised actor policies $\{\mu_1, \ldots, \mu_N\}$.

Initialise for each agent $i = 1, \ldots, N$:
    Actor network μ_i with parameters θ_i.
    Residual critic network $Q_{res,i}$ with parameters ϕ_i.
    Target networks μ'_i and $Q'_{res,i}$ with weights $\theta'_i \leftarrow \theta_i$ and $\phi'_i \leftarrow \phi_i$.
Initialise replay buffer $\mathscr{D}$ and Collaborative Global Model f_{sim}.

for episode = 1 to M **do**
    Receive initial observations $\{o_1, \ldots, o_N\}$.
    for t = 1 to max_steps **do**
        For each agent i, select action $a_i = \mu_i(o_i) + \mathcal{N}_t$.
        Let joint action be $a = \{a_1, \ldots, a_N\}$.
        Execute a, observe rewards $r = \{r_1, \ldots, r_N\}$ and next observations $\{o'_1, \ldots, o'_N\}$.
        Store the tuple $(\{o_i\}, \{a_i\}, \{r_i\}, \{o'_i\})_{i=1..N}$ in the replay buffer $\mathscr{D}$.
        if replay buffer $\mathscr{D}$ contains enough samples **then**
            Sample a random minibatch of K experiences.
            Let a sample k be $(\{o_i^k\}, \{a_i^k\}, \{r_i^k\}, \{o'_{i^k}\})$.
            ▷ Perform simulation rollout using joint information
            Calculate value baselines $R^k_{sim} = \text{Rollout}(f_{sim}, \{o_i^k\}, \{a_i^k\}, \text{horizon} = k')$.
            for agent $i = 1$ to N **do**
                ▷ Update critic for agent i using global information
                Form the target joint action: $a'^k = \{\mu'_1(o'^k_1), \ldots, \mu'_N(o'^k_N)\}$.
                Calculate target simulation baseline:
                $R^k_{sim,next} = Rollout(f_{sim}, \{o'^k_i\}, a'^k, \text{horizon} = k')$.
                Set target value: $y_i^k = r_i^k + \gamma(R^k_{sim,next} + Q'_{res,i}(\{o'^k_j\}_{j=1..N}, a'^k))$.
                Update critic by minimizing the loss:
                $\mathscr{L}(\phi_i) = \frac{1}{K}\sum_k \left(y_i^k - (R^k_{sim} + Q_{res,i}(\{o_j^k\}_{j=1..N}, \{a_j^k\}_{j=1..N}))\right)^2$.
                Update actor for agent i using the sampled policy gradient w.r.t Q_i.
            end for
            for agent $i = 1$ to N **do** ▷ Soft update all target networks
                $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$
                $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
            end for
        end if
    end for
end for

agents. This includes creating an **actor network** (μ_i) to decide actions and a **residual critic network** ($Q_{res,i}$) to learn the long-term strategic value. Corresponding **target networks** (μ'_i and $Q'_{res,i}$) are created and their weights are initialised to be identical to the main networks. A central **replay buffer** ($\mathscr{D}$) is set up to store experiences, and the **Collaborative Global Model** (f_{sim}) is constructed from the shared models of the agents, as described in Section 6.3.

2. Data Collection: The algorithm enters a loop of training episodes. Within each episode, the agents interact with the environment to collect data.

- **Action Selection:** Each agent i observes its local state o_i and selects an action a_i using its actor policy μ_i. To encourage exploration, random noise ($\mathscr{N}_t$) is added to the action.
- **Environment Interaction:** The joint action of all agents is executed in the environment, which transitions to a new state. The agents receive their respective rewards r_i and new local observations o'_i.
- **Experience Storage:** The complete transition tuple, containing the observations, actions, rewards, and next observations for all agents, is stored in the replay buffer $\mathscr{D}$.

3. Centralised Training: Once the replay buffer has accumulated enough experiences, the centralised training phase begins. A random minibatch of K experiences is sampled from the buffer for each training step.

- **Simulation Rollout and Baseline Calculation:** This is a key step unique to DT-MADDPG. For each experience in the minibatch, the Collaborative Global Model (f_{sim}) is used to perform a **simulation rollout** for a short horizon of k' steps. This simulation generates a short-term trajectory of rewards, which are summed to produce the physics-based value baseline, R_{sim}. This value represents the "known" part of the future reward, grounded in the system's physics.
- **Critic Update:** The residual critic network for each agent is updated.

  - First, the target actions a' are computed using the target actor networks.
  - Then, another simulation rollout is performed from the *next* state s' with these target actions to calculate the simulation baseline for the next state, $R_{sim,next}$.
  - The **target value** y_i is constructed. It is the sum of the immediate reward r_i and the discounted future value. Crucially, this future value is itself decomposed into the known physics part ($R_{sim,next}$) and the learned strategic part ($Q'_{res,i}$).
  - Finally, the residual critic network $Q_{res,i}$ is updated by minimizing the loss between this target value y_i and the current composite Q-value estimate, which is the sum of the current simulation baseline R_{sim} and the current residual value $Q_{res,i}$. This forces the critic to only learn the "error" or residual value that the simulator cannot predict.

- **Actor Update:** Each agent's actor network is updated using the policy gradient. The gradient is calculated with respect to the full, composite Q-value ($Q_i = R_{sim} + Q_{res,i}$), encouraging the actor to take actions that maximise both the short-term, physics-based rewards and the long-term, learned strategic value.
- **Target Network Updates:** To stabilise learning, the weights of the target actor and critic networks are slowly updated ("soft update") to track the weights of the main networks. This is a standard technique in deep Q-learning and its variants.

After many episodes, this process results in a set of trained, decentralised actor policies that can coordinate effectively using only local observations, having learned from the global perspective provided by the simulation-assisted critic during the offline training phase.

6.5 Experimental Setup

6.5.1 Simulation Environment

To validate the effectiveness of our proposed DT-MADDPG architecture, we designed a series of experiments within a simulated V2G environment. The objective is to assess the ability of our framework to learn effective coordination policies by comparing its performance on several key metrics against established baseline methods.

We developed a discrete-event based simulation representing a local power distribution grid servicing a community with a fleet of EVs. The simulator is programmed in Python using the SimPy library and the objects in the simulation such as EV, generators and consumers are modelled as SimPy processes. The environment consists of a power grid model and the EV fleet model. The grid has a fluctuating base load from residential and commercial uses and two primary power sources: a main grid connection representing non-renewable (fossil fuel) power, and a local renewable energy source (e.g., solar or wind farms) with variable output based on historical data. In these settings, we assume that the load of the grid can be balanced by adjusting the output power generated by non-renewable resources. A dynamic pricing mechanism dictates the cost of drawing power from and the revenue from selling power back to the main grid. The dynamic pricing is segmented, with three periods - Peak, Valley and Normal. The detailed price is listed in Table 6.2. The data is taken from the residential electricity prices in the city of Shenzhen, China. We also assume the price of selling electricity back to the grid is the same as the energy price in the given period.

Table 6.2: Dynamic Electricity Price used in the Simulation, from Shenzhen, China, in CNY/kWh.

Period	Time (24h)	Price (CNY/kWh)
Peak	10-12; 14-19	1.1121
Normal	8-10; 12-14; 19-24	0.6542
Valley	0-8	0.2486

In the simulation, a fleet of N EVs, each represented by a Digital Twin agent, is placed in the grid

environment. For the primary performance comparison experiments (grid stability, utilisation, and user satisfaction), we set the fleet size to $N = 50$. For the scalability analysis, N is varied from 10 to 200, as detailed in Section 6.6. Each EV is assigned a unique battery capacity, charging/discharging efficiency, and a stochastic behavioural profile governing its daily arrival time, departure time, and energy consumption. To study the scalability of the system, we have also constructed a model of the network infrastructure. While there are no constraints on the bandwidth or latency, it is still worth studying the communication overhead of different approaches. For the network model, we used the Barabási-Albert model [12]. This model was chosen because it generates scale-free networks, which accurately represent the topology of many real-world complex systems, including infrastructure and communication networks. Scale-free networks are characterised by a power-law degree distribution, meaning they contain a small number of highly connected "hubs" and a large number of nodes with fewer connections. In the context of a V2G network, these hubs can be seen as analogous to large public charging stations or key grid substations. Using the Barabási-Albert model provides a more realistic and challenging topology for evaluating communication overhead, as it allows us to test the robustness of our decentralised protocol against the natural formation of communication bottlenecks at these hubs. The experiments are conducted on a bare-metal server with 4 Intel Xeon Gold 6230 CPU and a NVIDIA RTX A6000 GPU.

6.5.2 Hyperparameter and Training Configuration

The neural networks for the actor and critic policies were trained using the Adam optimiser. The key hyperparameters for the RL algorithms and network architectures are detailed in Table 6.3. These values were selected based on common practices in the literature and empirical tuning to ensure stable convergence across all baseline methods. The actor network uses a *tanh* activation function for its final layer to constrain the output action to the range [-1, 1], which is then scaled to the appropriate power flow limits.

Table 6.3: Hyperparameters for MARL Training, including DT-MADDPG and Baselines.

Parameter	Value
RL Algorithm Parameters	
Discount Factor (γ)	0.99
Soft Update Rate (τ)	0.005
Replay Buffer Size	1,000,000
Batch Size	256
Simulation Rollout Horizon (k')	5 steps
Neural Network Architecture	
Actor Network	2 hidden layers, [128, 128] neurons
Residual Critic Network	2 hidden layers, [128, 128] neurons
Hidden Layer Activation	ReLU
Actor Output Activation	tanh
Optimiser Parameters	
Optimiser	Adam
Actor Learning Rate	1×10^{-4}
Critic Learning Rate	1×10^{-3}

Justification of Hyperparameters

The selection of hyperparameters is critical for the successful training of deep reinforcement learning agents. The values presented in Table 6.3 were chosen based on a combination of established best practices from the literature and empirical tuning specific to the V2G domain.

RL Algorithm Parameters

- **Discount Factor ($\gamma = 0.99$):** A high discount factor is standard for episodic tasks with long horizons. It encourages the agents to adopt far-sighted policies, which is essential in V2G coordination where current charging decisions have long-term consequences on meeting future departure targets [116].
- **Soft Update Rate ($\tau = 0.005$):** This small value ensures that the target networks update slowly, providing a stable learning target for the actor and critic networks. This is a common practice

in DDPG-based algorithms to prevent learning oscillations and divergence [74].

- **Replay Buffer Size (10^6):** A large replay buffer is crucial for breaking temporal correlations in the collected experiences, a key requirement for off-policy learning. A size of one million ensures a diverse set of past experiences is available for minibatch sampling, improving learning stability [87].
- **Simulation Rollout Horizon ($k' = 5$):** This parameter, unique to DT-MADDPG, represents a trade-off. The horizon must be long enough to capture the immediate physical consequences of an action but short enough to ensure the simulation remains accurate and computationally tractable. A 5-step horizon (representing 1.25 hours in our 15-minute timestep simulation) was found empirically to provide a robust, physics-based value baseline without introducing excessive computational overhead during training.

Network Architecture and Optimiser

- **Network Size ([128, 128]):** A two-layer network with 128 neurons per layer provides sufficient capacity to approximate the policy and value functions for this control problem without being overly complex, which could lead to overfitting. This architecture is a common baseline for continuous control tasks [79].
- **Optimiser (Adam):** The Adam optimiser was selected due to its widespread adoption and proven effectiveness in deep reinforcement learning applications [64]. Adam combines the advantages of two other extensions of stochastic gradient descent: AdaGrad, which handles sparse gradients well, and RMSProp, which works well in non-stationary settings. Its adaptive learning rate mechanism, which computes individual learning rates for different parameters, makes it particularly well-suited for training the large, complex neural networks used in this research, providing robust and efficient convergence.
- **Learning Rates (10^{-4} for Actor, 10^{-3} for Critic):** It is standard practice in actor-critic meth-

ods to use a smaller learning rate for the actor than the critic. The critic's value estimates are often noisy, so a higher learning rate helps it converge faster. The policy update, however, is more sensitive, and a lower learning rate prevents large, destabilising policy changes [74].

- **Activation Functions (ReLU, tanh):** ReLU is used for hidden layers due to its computational efficiency and ability to mitigate vanishing gradients. The *tanh* activation on the actor's output layer is essential for constraining the action to a normalised range of [-1, 1], which is then scaled to the environment's specific action space.

6.5.3 Methodology and Statistical Considerations

To ensure the robustness and statistical validity of our findings, the experimental methodology incorporates several considerations for handling the non-determinism inherent in both the simulation environment and the reinforcement learning process.

Sources of Non-Determinism: The simulation environment includes multiple sources of stochasticity to reflect real-world conditions. Each EV agent is initialised with a stochastic behavioural profile, including random arrival times, departure times, and initial States of Charge, drawn from predefined distributions. Furthermore, the renewable energy generation and base grid load follow patterns derived from historical data, which include inherent variability. On the algorithmic side, the MARL training process itself introduces randomness through the exploration noise added to actions and the random sampling of experiences from the replay buffer.

Variance and Repeatability: To account for this variability, each experiment was executed 10 times with different random seeds. The results presented in the subsequent figures (e.g., Figure 6.2) depict the **mean performance** across these runs. The variance across runs was observed to be low, indicating stable and repeatable behaviour from the learning algorithms. For clarity in the line plots, only the mean is shown, but the low variance confirms that the observed performance differences between the algorithms are statistically significant and not merely artefacts of random chance.

Sensitivity Analysis: A sensitivity analysis was conducted to understand the robustness of the algorithms to changes in key parameters. The analysis presented in Section 6.6 (Figures 6.4 and 6.5), which varies the user's revenue preference weight ($w_{revenue}$), serves as a direct sensitivity analysis of the reward structure. This demonstrates how the algorithms adapt to different user priorities. Furthermore, the hyperparameters listed in Table 6.3 were selected after a process of empirical tuning, which implicitly tests the sensitivity of the models to parameters like learning rates and network architecture. Finally, the scalability analysis, where the number of agents (N) is varied, acts as a sensitivity analysis with respect to the scale of the system.

6.5.4 Evaluation Metrics

We evaluate the performance of all approaches using a combination of scenario-specific and system-level metrics:

- Grid Stability: The primary goal of V2G is to stabilise the grid by reducing reliance on fossil fuels during peak demand (peak shaving) and absorbing excess energy during off-peak times (valley filling). We visualise this by plotting the power drawn from the non-renewable source over time.
- Renewable Energy Utilisation: This metric measures the percentage of available renewable energy that is successfully used by the system, either consumed directly or stored in EV batteries and consumed in the future, instead of being disposed of. This will be presented as a bar chart, where higher utilisation is desirable.
- User Satisfaction: We measure the framework's ability to meet the needs of individual EV owners via the Owner Goal Success Rate. This is the percentage of EVs that successfully meet their owner-defined State-of-Charge (SoC) requirements by their scheduled departure time. Results will be compared across algorithms using a bar chart.

- User Revenue: To assess the economic benefit for participants, we calculate the Aggregated User Revenue. This is the total income earned by all EV owners from selling energy back to the grid, minus their total charging costs. This will also be presented in a comparative bar chart.
- Communication Overhead: The efficiency of the underlying collaborative framework is measured by the average number of messages exchanged between DTs per simulation step. Also, to showcase the potential network congestion, we plot the distribution of network load (average number of messages flowing through a network node) to reveal the network usage.

6.5.5 Baselines

We evaluate the performance of our proposed DT-MADDPG algorithm against two distinct MARL baselines to demonstrate its advantages:

- Independent Learners (IL): In this approach, each EV agent is a fully decentralised reinforcement learner with its own actor and critic. It learns a policy based solely on its local observations, representing a non-collaborating but still intelligent baseline.
- Standard MADDPG (MADDPG): This is the canonical centralised training with decentralised execution algorithm. A central critic collects state and action information from all agents to guide training. This baseline is the crucial ablation study, allowing us to isolate and measure the performance gain attributable directly to our novel simulation-assisted critic.

6.6 Results and Analysis

6.6.1 Grid Stability and Renewable Utilisation

First, we assessed the core performance of the models in managing the power grid. Figure 6.2 illustrates grid stability by plotting the required fossil fuel generation (Y-axis) over a 24-hour period

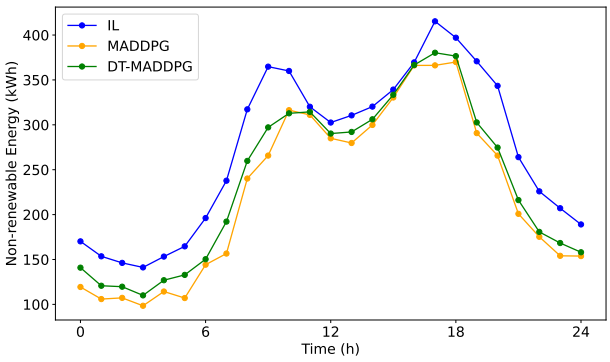

Figure 6.2: Energy drained from non-renewable energy sources, indicating grid stability.

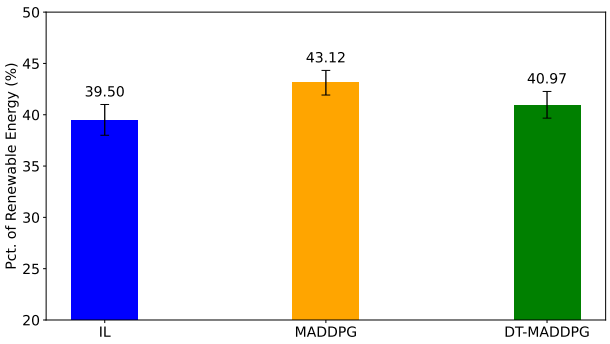

Figure 6.3: Percentage of renewable energy utilisation with different algorithms.

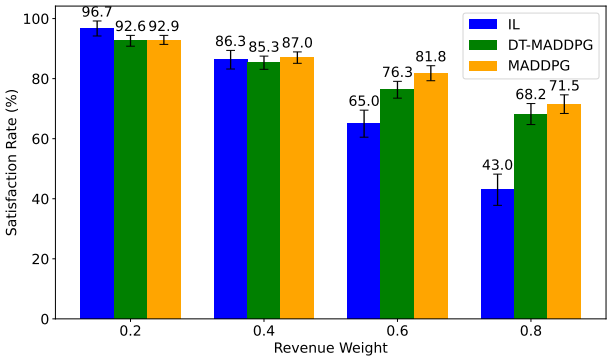

Figure 6.4: User satisfaction rate as a function of the revenue preference weight ($w_{revenue}$).

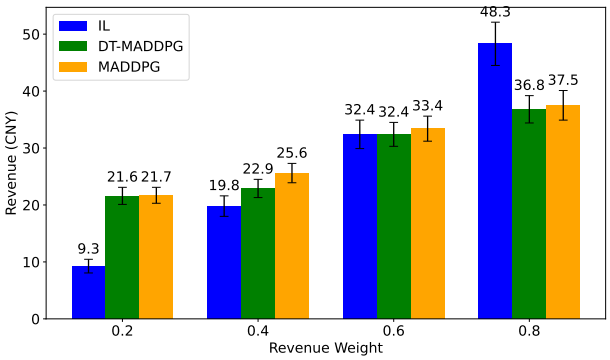

Figure 6.5: Aggregated user revenue as a function of the revenue preference weight ($w_{revenue}$)

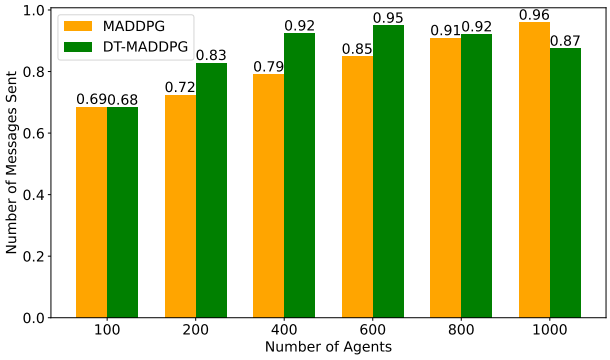

Figure 6.6: Scalability of communication overhead versus the number of agents.

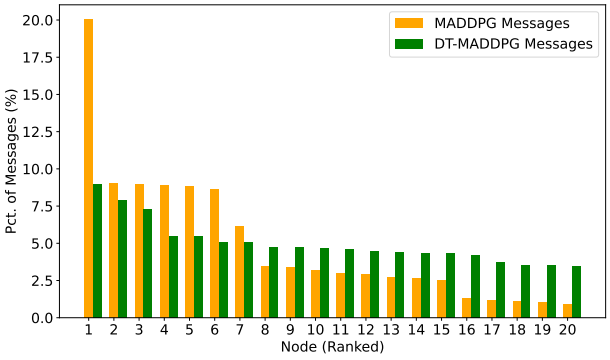

Figure 6.7: Distribution of network communication load across nodes.

(X-axis). The results clearly show that both MADDPG and DT-MADDPG outperform the IL baseline by requiring less fossil fuel. Crucially, DT-MADDPG achieves stability comparable to the fully centralised MADDPG, proving its effectiveness in coordinating the V2G network.

This high performance extends to the use of renewables, as shown in Figure 6.3. DT-MADDPG achieves a renewable energy utilisation of 40.97%, a result that is nearly identical to MADDPG and significantly higher than IL. These findings establish that DT-MADDPG maintains top-tier performance on primary system objectives without requiring full centralisation.

6.6.2 Multi-Objective Robustness

To investigate how the models adapt to different user priorities, we conducted a sensitivity analysis by varying the revenue weight, $w_{revenue}$, from 0.2 to 0.8. A higher $w_{revenue}$ encourages agents to prioritise revenue generation over maintaining their target State of Charge. As expected, increasing this weight leads to a clear trade-off: user satisfaction declines (Figure 6.4) while user revenue increases (Figure 6.5) for all methods. However, a key trend emerges: the IL baseline is highly sensitive to this parameter change, showing volatile swings in performance. In contrast, MADDPG and DT-MADDPG demonstrate significantly more stability. This resilience can be attributed to their coordination mechanisms; while IL agents narrowly focus on local, weight-sensitive goals, MADDPG and DT-MADDPG agents must balance these with the global objective of grid stability. This moderates their response to changes in local incentives, leading to more robust and predictable system behaviour.

6.6.3 Impact of Simulation Horizon

To evaluate the trade-off between foresight capability and model reliability, we analyse the impact of the simulation rollout horizon (k') on system performance. This analysis is crucial because the core innovation of DT-MADDPG relies on the Collaborative Global Model, which is subject to simulation

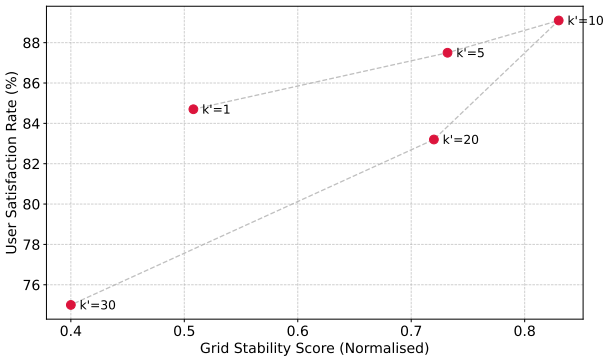

Figure 6.8: Pareto frontier of Grid Stability Score vs User Satisfaction Rate across varying simulation rollout horizons (k').

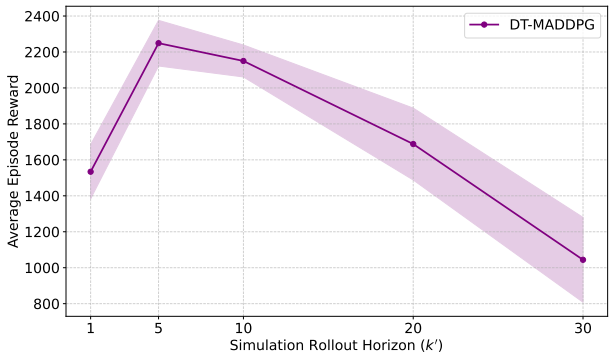

Figure 6.9: Average Episode Reward as a function of simulation rollout horizon (k').

error accumulation over time.

Trade-off between Stability and Satisfaction

Figure 6.8 illustrates the Pareto frontier of the Grid Stability Score against the User Satisfaction Rate. The results demonstrate a clear performance improvement as the rollout horizon increases from $k' = 1$ to $k' = 10$. In this range, the system exhibits a "win-win" trajectory where both objectives improve simultaneously. Specifically, compared to the short-sighted baseline of $k' = 1$, the agent with a horizon of $k' = 10$ achieves a significant increase in the normalised Grid Stability Score (from ≈ 0.51 to ≈ 0.83) while concurrently boosting the User Satisfaction Rate from 84.7% to 89.1%. This suggests that a moderate look-ahead window allows agents to effectively anticipate future grid peaks and user departure constraints.

However, extending the horizon beyond this optimal point ($k' > 10$) leads to a sharp degradation in performance. As observed at $k' = 20$ and $k' = 30$, the system retreats from the Pareto frontier. At $k' = 30$, the grid stability score collapses to 0.40, and user satisfaction drops to 75.0%. This phenomenon is attributed to *simulation model drift*, where small approximation errors in the system dynamics compound recursively over longer rollouts, causing the critic to optimise based on a divergent future

trajectory.

Optimisation of the Planning Window

To determine the optimal planning window, we further analysed the Average Episode Reward, as shown in Figure 6.9. The results reveal a distinct inverted U-shaped relationship. The initial extension of the horizon from a short-sighted setting ($k' = 1$) to a short-term planning window ($k' = 5$) yields the most significant performance gain, with the average reward surging from approximately 1550 to a peak of ≈ 2250.

Conversely, extending the horizon beyond $k' = 10$ results in a monotonic decrease in performance. At $k' = 30$, the average reward falls to ≈ 1050, performing worse than the baseline agent with no planning capability. Furthermore, the widening error bands observed at longer horizons highlight the growing instability of the learning process. These findings empirically validate that a moderate horizon of $k' \in [5, 10]$ represents the "sweet spot" for this architecture, balancing the strategic benefits of planning against the accumulation of simulation error.

6.6.4 Scalability and Network Load

A key claim of this thesis is that the proposed decentralised framework offers superior scalability compared to traditional centralised approaches. This section analyses the communication overhead and network load distribution to validate this claim.

Analysis of Total Communication Volume

Figure 6.6 shows the total number of messages exchanged per time-step as the number of agents (N) in the network increases. Both the centralised MADDPG and the decentralised DT-MADDPG exhibit a linear scaling, $O(N)$, in total communication volume. This is expected:

- In **MADDPG**, each of the N agents sends its state and action to the central critic, resulting in

N messages per time-step.

- In **DT-MADDPG**, each of the N agents broadcasts its decision information to its local peers. Assuming each broadcast is a single network event, this also results in N messages per time-step.

While the total message count is comparable, this metric alone is misleading. The critical difference, and the key to true scalability, lies not in the total volume of traffic but in its distribution across the network.

Analysis of Network Load Distribution and Bottlenecks

Figure 6.7 provides a histogram of the communication load (number of messages received) per node, revealing the fundamental architectural differences between the two approaches.

Centralised Bottleneck in MADDPG In the MADDPG architecture, all N messages converge on a single destination: the centralised critic. This creates an extreme imbalance in network load. As shown in the figure, one node (the critic) receives a number of messages proportional to the entire network size, $O(N)$, while all other nodes (the actors) receive none. This central node becomes a severe **computational and network bottleneck**. As N increases, the critic's capacity to process incoming messages and compute value updates will inevitably be saturated, placing a hard limit on the scalability of the entire system. Furthermore, this design creates a single point of failure; the loss of the critic node would halt the learning process for all agents.

Distributed Load in DT-MADDPG In contrast, the DT-MADDPG framework distributes the communication load across the network. Messages are exchanged in a peer-to-peer fashion, so the load on any individual node is determined by the size of its local peer group, not the total network population. The histogram for DT-MADDPG shows a much flatter and more balanced distribution. While the use of a Barabási-Albert network model means some "hub" nodes have higher connectivity and thus a

higher message load, this load is orders of magnitude smaller than that of the MADDPG critic. The load on any given node is $O(p_i)$, where p_i is the number of its peers. Since the average peer count in a scale-free network remains small even as N grows, the average load per node remains manageable.

Synthesis: Scalability and Robustness

The analysis demonstrates a clear architectural advantage for DT-MADDPG. While both systems have a total communication cost that scales linearly with the number of agents, MADDPG concentrates this cost onto a single node, creating a bottleneck that prevents true scalability. DT-MADDPG, by distributing this cost, ensures that no single node is overwhelmed. This decentralised communication pattern not only enables the system to scale to much larger numbers of agents but also significantly improves its robustness by eliminating the single point of failure inherent in the centralised design. This confirms that the proposed architecture successfully meets the scalability and robustness requirements for real-world deployment in Large Complex Systems.

6.7 Discussion

The experimental results validate the core hypothesis of this thesis: it is possible to achieve intelligent coordination in decentralised systems without compromising privacy. The DT-MADDPG algorithm bridges the gap between the "privacy" of decentralised architectures and the "performance" of centralised training. However, to fully understand the implications of this work, it is necessary to critically analyse its current limitations and the path toward even more rigorous privacy guarantees.

6.7.1 Limitations of the Current Approach

While the framework successfully enables coordination, several specific limitations must be acknowledged:

Fidelity of the Simulation-Assisted Critic and the Sim-to-Real Gap

The core innovation of DT-MADDPG, the simulation-assisted critic, is also its primary source of potential limitations. The algorithm's performance is fundamentally predicated on the assumption that the *Collaborative Global Model* is a high-fidelity proxy for the physical world. Any divergence between the simulation and reality introduces challenges related to the rollout horizon, critic bias, and the broader "Sim-to-Real" gap.

The Rollout Horizon (k') Trade-off The choice of the simulation rollout horizon, k', represents a critical trade-off. A short horizon is computationally efficient and minimises the accumulation of simulation errors, but the resulting baseline R_{sim} may be too simplistic to provide a meaningful learning signal. Conversely, a long horizon can capture more complex, near-term dynamics but is computationally expensive and risks amplifying small model inaccuracies over the rollout. This can lead to a highly biased R_{sim} that misleads the learning process. The optimal value for k' is therefore domain-dependent and a sensitive hyperparameter.

The Problem of Critic Bias The value decomposition strategy trains the residual critic, $Q_{res,i}$, to learn the long-term value that the simulation rollout cannot capture. However, if the simulation baseline R_{sim} is systematically biased due to model inaccuracies, the critic will be forced to learn a correction for this *model error* rather than learning the true, underlying strategic value of a state. This can lead to a policy that is overfitted to the specific flaws of the simulator. Such a policy might perform well in the biased training environment but fail to generalise to the real world, exacerbating the Sim-to-Real gap.

Sources of Model Inaccuracy The fidelity of the Collaborative Global Model can be compromised by several factors:

- **Systemic Bias:** The shared environment model (e.g., grid physics, base load profiles) is pro-

vided by the system operator and may contain simplifying assumptions or be based on historical data that does not fully capture the dynamics of all local sub-systems, introducing a systemic bias.

- **Contribution Bias:** The framework assumes that when agents register their internal models (e.g., battery efficiency curves), these models are accurate. In a competitive environment, there is an incentive for agents to provide biased models. For example, an EV manufacturer might submit an overly optimistic battery model to make its vehicles appear more capable in the V2G market. This "contribution bias" would directly corrupt the Collaborative Global Model.
- **Unmodeled Dynamics:** Factors such as hardware variability, sensor noise, and complex battery degradation physics can create a discrepancy between the simulated baseline (R_{sim}) and reality.

If the Collaborative Global Model is significantly biased, the "physics-based" component of the value decomposition becomes unreliable, potentially providing a misleading learning signal and destabilizing the training process. While the trust mechanism from Chapter 5 can identify peers with inaccurate *predictions*, it does not directly audit the fidelity of the underlying *models* they contribute to the global simulator. Developing mechanisms for "model verification" or learning to correct for simulation bias are critical avenues for future work.

Assumption of Homogeneity and Reward Stability

Our current experimental setup assumes relatively homogeneous user profiles and preferences. Real-world V2G systems involve extreme heterogeneity, from commercial fleets with strict schedules to private users with erratic behaviour. Such diversity could introduce complex, non-linear coordination dynamics. Furthermore, the use of multi-objective reward functions, especially with heterogeneous agent preferences, can lead to learning instability. When agents pursue conflicting goals (e.g., maximizing personal revenue versus ensuring global grid stability), their learning updates can work against each other, leading to oscillatory or suboptimal convergence.

Several strategies could be employed to mitigate this instability. **Reward shaping**, particularly Potential-Based Reward Shaping (PBRS), could provide auxiliary rewards to guide agents toward cooperative equilibria without altering the underlying optimal policy. Another approach is **curriculum learning**, where agents initially train on a simplified objective (e.g., only SoC satisfaction) and are gradually exposed to more complex, conflicting objectives as their policies mature. Finally, more advanced **multi-objective MARL** frameworks could be adopted, which aim to learn a Pareto front of policies, allowing a system operator or user to select the desired operating point from a set of optimal trade-offs, rather than relying on pre-defined scalar weights.

Lack of Formal Privacy Quantification

While our framework is *architecturally* privacy-preserving (by not transmitting raw data), we have not yet provided a formal, quantitative analysis of privacy leakage . It is theoretically possible that sophisticated adversarial attacks (e.g., model inversion) could infer sensitive attributes from the broadcasted decision streams. The current work treats privacy as a binary constraint (share vs. don't share) rather than a continuous variable to be optimised.

Computational Cost of Simulation-Assisted Learning

The DT-MADDPG algorithm introduces a "Simulation Rollout" step that is computationally more intensive than the critic update in standard MADDPG. For each experience tuple in a training batch, the Collaborative Global Model must be executed for a rollout horizon of k' steps. This is a deliberate design trade-off: we accept a higher computational cost during the offline training phase in exchange for improved sample efficiency, faster convergence, and a more stable learning signal, as the simulation provides a rich, physics-grounded value baseline.

However, this cost is manageable due to the nature of simulation. The Collaborative Global Model is a purely computational artifact and can be executed in a **faster-than-real-time** manner, especially

when parallelised on hardware like GPUs. The training process is therefore not constrained by the wall-clock time of the physical system it represents. Future work could further mitigate this cost by exploring adaptive rollout horizons, where k' is dynamically adjusted based on the stability of the system state, or by using lightweight surrogate models to approximate the full simulator for less critical training steps.

6.7.2 Future Outlook: Quantifying Privacy in the Reward Function

Addressing the third limitation represents the most promising avenue for future research: moving from "Privacy by Architecture" to "Privacy by Optimisation."

Currently, agents optimise for Grid Stability (R_{global}) and Revenue (R_{local}). We propose extending this to explicitly treat Privacy as a quantifiable objective within the reward function. This would transform the problem into a multi-objective optimisation task where agents learn to trade off performance for privacy.

Formalisation

We can define a privacy loss metric, L_{priv}, which quantifies the information leakage of a shared action or prediction. For example, using Differential Privacy concepts, L_{priv} could measure the reduction in uncertainty an adversary gains about the agent's private state s_i given the shared information I_{shared}:

$$L_{priv}(t) = H(s_i) - H(s_i|I_{shared}(t)) \tag{6.10}$$

where H represents entropy.

We can then augment the agent's reward function $r_i(t)$ with a penalty term:

$$r'_i(t) = r_i(t) - \lambda_{priv} \cdot L_{priv}(t) \tag{6.11}$$

where λ_{priv} is a dynamic coefficient representing the user's sensitivity to privacy.

Expected Outcome

By incorporating this term, the DT-MADDPG agents would be forced to learn a Pareto-optimal frontier. They might discover strategies such as "noise injection" or "temporal aggregation", intentionally broadcasting slightly vague or delayed predictions to minimise L_{priv}, while still providing enough useful signal to maintain grid stability. This would allow the system to mathematically guarantee a specific level of privacy utility, customizing the behavior for each individual user's risk tolerance.

6.8 Summary

This chapter presented the **DT-MADDPG** algorithm, completing the "Intelligence" layer of the proposed framework. By integrating the trust-verified decision sharing (Chapter 5) with a novel simulation-assisted learning mechanism, we enabled autonomous Digital Twin agents to learn complex, coordinated strategies. The results demonstrate that this approach achieves near-optimal grid stability and renewable utilisation while strictly preserving data privacy and ensuring scalable, bottleneck-free communication.

Chapter 7

Conclusion and Future Work

7.1 Summary of Work

The central objective of this thesis was to address the inherent conflict between the need for system-wide optimisation in Large Complex Systems and the constraints of privacy, scalability, and decentralisation required for their physical deployment. Using Smart Energy Grids and Vehicle-to-Grid networks as the primary application domain, this research investigated how autonomous Digital Twins can effectively cooperate and coordinate without relying on a central authority or exposing sensitive raw data.

The thesis proceeded through several logical stages to build a comprehensive solution. Initially, we defined the problem landscape (Chapters 1, 2, and 3), identifying that traditional centralised Digital Twin architectures suffer from critical bottlenecks in scalability and privacy, while existing decentralised models lack the mechanisms for intelligent coordination. To bridge this gap, Chapter 4 established a novel peer-to-peer middleware that enables agents to exchange high-level decision information (intentions) rather than raw data, providing the foundational infrastructure for privacy-preserving cooperation. Acknowledging the uncertainty inherent in open systems, Chapter 5 introduced a Trust

Mechanism that mathematically verifies peer predictions against physical reality, allowing agents to dynamically filter unreliable peers and maintain fault tolerance. Finally, Chapter 6 addressed the optimisation challenge by introducing the DT-MADDPG algorithm, a hybrid Multi-Agent Reinforcement Learning approach that leverages a Collaborative Global Model to enable agents to learn globally optimal strategies without ever accessing private data.

7.2 Key Findings

The experimental evaluations conducted throughout this thesis yield three primary findings that validate the proposed methodology:

First, we demonstrated that **Decision-Level Sharing enables effective cooperation**. The framework design proved that exchanging raw data is not a prerequisite for coordination. In the VPP scenarios (Chapter 5), agents sharing only abstract "intentions" (power profiles) achieved a significantly faster response to grid demands, reaching the target power output in 105 seconds compared to 175 seconds for the independent baseline, a 40% improvement in agility. This confirms that high-level decision information contains sufficient signal to coordinate complex physical actions while acting as an effective privacy barrier.

Second, our results highlight that **Trust Mechanisms are essential for robustness**. In decentralised networks, blindly trusting peer information is a severe vulnerability. Experiments demonstrated that a single faulty peer could destabilise the grid by misleading other agents. However, the proposed Trust Mechanism allowed the system to dynamically identify and isolate unreliable agents based on historical accuracy, maintaining grid stability where standard collaborative approaches failed.

Third, we established that **Simulation-Assisted Learning bridges the privacy-performance gap**. The DT-MADDPG algorithm achieved grid stability and renewable energy utilisation rates (40.97%) comparable to the fully centralised, non-private MADDPG baseline (43.12%) and significantly supe-

rior to the Independent Learner baseline (39.50%). This confirms that a Collaborative Global Model built from privacy-preserving signals provides a sufficiently accurate proxy environment for training sophisticated AI agents without exposing sensitive user data.

7.3 Research Contributions

This thesis contributes to the fields of Distributed Systems and Multi-Agent Reinforcement Learning by bridging the gap between architectural privacy and algorithmic intelligence. The specific contributions are as follows:

1. **A Decentralised Digital Twin Middleware:** We proposed a novel architectural framework that decouples the internal state execution of a Digital Twin from its external information exchange. By standardizing a "Decision-Level" communication protocol, this middleware enables heterogeneous agents (e.g., EVs, Batteries, Turbines) to collaborate effectively without the need for raw data interoperability or centralised control.

2. **A "Decision-Level" Trust Protocol:** We developed a mathematical mechanism for verifying the reliability of high-level predictions in open systems. Unlike traditional security models that rely on identity verification, this protocol verifies *behavioural accuracy* by mapping historical predictions to ground-truth sensor data, ensuring secure collaboration even in the presence of faulty or malicious peers.

3. **The DT-MADDPG Algorithm:** We introduced a novel Multi-Agent Reinforcement Learning algorithm that incorporates a "Simulation-Assisted Critic". This contribution advances the Centralised Training with Decentralised Execution paradigm by demonstrating how physics-based simulations can replace raw data aggregation, allowing for the training of sophisticated, globally optimised policies while strictly adhering to privacy constraints.

7.4 Broader Impact and Generalisability

While this thesis has focused on the V2G domain, the core principles of the proposed framework, the Decision-Level Sharing, Trust Verification, and Simulation-Assisted Learning, are designed to be domain-agnostic. They provide a generalisable template for enabling collaboration in a wide range of multi-owner Large Complex Systems where the Privacy-Coordination Paradox is a central challenge.

In **Smart Urban Logistics**, a network of autonomous delivery vehicles or drones operated by competing companies could coordinate by sharing abstracted decision-level information (e.g., predicted arrival times or intended charging hub usage) to deconflict traffic and schedule resources without exposing proprietary route data or customer manifests. Similarly, in **Smart Manufacturing (Industry 4.0)**, heterogeneous robotic systems from different vendors can negotiate resource access and optimise production flow by sharing intended path segments or machine utilisation schedules, keeping sensitive production recipes and tool-wear statistics strictly private. Furthermore, in **Distributed Healthcare Monitoring**, a network of wearable medical devices could collaborate to build holistic patient models by sharing high-level risk scores or anomaly flags (e.g., a predicted hypoglycemic event), allowing for early warning and coordinated intervention while ensuring raw physiological data remains secure and local.

In each of these domains, the framework provides a robust solution to achieving the global awareness needed for optimisation while respecting the local data sovereignty required for participation. The V2G case study, with its extreme volatility and strict privacy needs, serves as a rigorous proof-of-concept, demonstrating the viability of this approach for a broad class of decentralised socio-technical systems.

7.4.1 Applicability and Limitations of the Framework

To contextualise the utility of the proposed framework, it is crucial to explicitly define the types of problems it is best suited for, its inherent preconditions, and scenarios where its application may be suboptimal or inappropriate.

Suitable Problem Characteristics

The framework is optimally suited for problems exhibiting the following characteristics:

- **Multi-Owner, Multi-Stakeholder Systems:** Problems where individual agents are owned by different entities with competing or conflicting objectives, necessitating a balance between local autonomy and global coordination (e.g., smart energy, urban logistics).
- **Privacy-Sensitive Data:** Domains where raw operational data is highly sensitive (e.g., personal schedules, proprietary processes) and cannot be centralised due to regulatory, ethical, or commercial constraints.
- **Large-Scale and Decentralised:** Systems comprising a large number of geographically distributed, heterogeneous assets (thousands to millions), where centralisation leads to scalability bottlenecks, single points of failure, or high latency.
- **Need for Real-time Coordination:** Environments requiring dynamic, adaptive responses to emergent conditions (e.g., sudden demand fluctuations, asset failures) that cannot be handled by static, pre-programmed rules.
- **Observable Physical Interactions:** Scenarios where agents' physical actions can be objectively verified by peers, even if only in aggregate, allowing the trust mechanism to function.
- **Simulation-Tractable Dynamics:** Problems where high-fidelity simulation models of individual assets and the aggregate environment are feasible, enabling the "what-if" analysis of Digital Twins and the "Simulation-Assisted Critic" of DT-MADDPG.

Preconditions for Framework Success

For the framework to be effectively deployed, certain preconditions must be met:

- **Agreed-Upon Ontology of Interfacing States:** All participating agents must agree *a priori* on the standardized format and semantics of the "Decision Information" (I_{dec}) that will be exchanged (e.g., power profiles, estimated arrival times). This common language is fundamental for Semantic Alignment.
- **Availability of Local High-Fidelity Models:** Each Digital Twin must possess an accurate internal model of its physical counterpart to generate reliable predictions and decisions locally. The quality of these local models directly impacts the efficacy of both coordination and trust verification.
- **Shared Observability (Overlap) for Verification:** As detailed in Section 5.3.5, for the Trust Mechanism to function, there must be a mechanism for one agent to objectively verify another's past predictions. This requires either direct or aggregated physical observability of the predicted state by the verifying agent.
- **Incentives for Cooperation:** While the framework lowers privacy barriers, a fundamental incentive structure must exist for agents to participate and adhere to coordinated actions. This can be implicit (e.g., avoiding grid collapse) or explicit (e.g., financial rewards for grid services).
- **Computational Capacity at the Edge:** Each Digital Twin node (or its associated edge device) must have sufficient local computational power to run its internal simulation, execute the Decision Maker, and manage the Middleware components.

Inapplicable Scenarios and Limitations

The framework is not a universal solution and may be less suitable under the following conditions:

- **Lack of Shared Physical Environment (Zero Overlap):** If agents operate in completely dis-

joint physical domains with no shared observable impact, the objective trust mechanism cannot function, as verification becomes impossible (as discussed in Section 5.3.5).

- **Highly Adversarial Environments with Unauditable Intentions:** While robust against faulty peers and strategic misinformation, the framework's privacy-preserving nature makes it challenging to detect and attribute intent in highly adversarial scenarios where agents might collude to deliberately misrepresent their *internal models* or inject subtle, coordinated biases that are not immediately verifiable from observable outputs.
- **Irreducible Raw Data Requirement:** If the problem inherently requires the aggregation of sensitive raw data for effective coordination (e.g., complex medical diagnoses requiring full patient histories across multiple institutions), then the privacy-preserving decision-level abstraction may be insufficient.
- **Resource-Constrained "Dumb" Devices:** For extremely lightweight IoT devices that lack the computational resources for local simulation, decision-making, and middleware execution, the overhead of running a full Digital Twin agent might be prohibitive.
- **Absence of Learning/Adaptation Need:** In static or simple environments where optimal rules can be hard-coded and remain effective indefinitely, the overhead of a learning-based, adaptive framework may outweigh the benefits.
- **Strong Central Authority Preferred:** In contexts where a single, trusted entity has the legal and social mandate for absolute central control and full data access, and scalability/robustness are not primary concerns, a traditional centralised approach might be simpler to implement.

7.5 Limitations and Future Work

While this thesis offers a comprehensive framework for decentralised coordination, several limitations inherent in the current scope open promising avenues for future research. A primary technical challenge is bridging the **Sim-to-Real Gap**. The validation of the DT-MADDPG algorithm relies on the fidelity of the Collaborative Global Model. Unmodeled physical discrepancies, such as battery degradation physics or communication latency, may cause the simulated training environment to diverge from reality. Future work should focus on validating the learned policies on physical hardware testbeds or through Hardware-in-the-Loop (HIL) simulations.

Furthermore, the current framework treats privacy as a binary architectural constraint. Future research will explore **Privacy-Regularised Reinforcement Learning**, incorporating a formal privacy loss metric (e.g., Differential Privacy) directly into the agent's reward function. This would allow the system to optimise an explicit trade-off between coordination performance and data disclosure. Additionally, expanding the environment to include **Heterogeneity and Economic Complexity** such as diverse user risk profiles and peer-to-peer energy auctions would allow the framework to move beyond cooperative load balancing into a true digital marketplace with emergent economic behaviours.

Beyond technical constraints, the deployment of such autonomous systems raises critical **Ethical Considerations and Societal Impacts**. Although the architecture prevents raw data sharing, aggregated decision-level data can still reveal sensitive patterns over time (e.g., inferring commuting routines from periodic demand drops), necessitating advanced obfuscation techniques like automatic noise injection. Moreover, the MARL algorithm risks learning policies that inadvertently introduce algorithmic bias, potentially favouring users with predictable schedules over those with flexible ones. Addressing this requires integrating fairness metrics into the learning process through constrained MARL or fairness-aware reward shaping [55, 51]. Finally, the "black-box" nature of neural network policies poses a significant challenge for transparency and accountability. Developing Explainable AI (XAI) [29] methods tailored for MARL, such as counterfactual explanations or saliency maps

[88], will be essential to provide audit trails and empower users with a clearer understanding of their agents' behaviour.

References

[1] Maheed H. Ahmed and Mahsa Ghasemi. “Privacy-Preserving Decentralized Actor-Critic for Cooperative Multi-Agent Reinforcement Learning”. In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li. Vol. 238. Proceedings of Machine Learning Research. PMLR, May 2024, pp. 2755–2763.

[2] Mudhafar Al-Saadi, Maher Al-Greer, and Michael Short. “Reinforcement Learning-Based Intelligent Control Strategies for Optimal Power Management in Advanced Power Distribution Systems: A Survey”. In: *Energies* 16.4 (2023). DOI: 10.3390/en16041608.

[3] Sally Aladdin, Samah El-Tantawy, Mostafa M. Fouda, and Adly S. Tag Eldien. “MARLA-SG: Multi-Agent Reinforcement Learning Algorithm for Efficient Demand Response in Smart Grid”. In: *IEEE Access* 8 (2020), pp. 210626–210639. DOI: 10.1109/ACCESS.2020.3038863.

[4] Kazi Masudul Alam and Abdulmotaleb El Saddik. “C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems”. In: *IEEE access* 5 (2017), pp. 2050–2062.

[5] Christopher Amato. “An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning”. In: *arXiv preprint arXiv:2409.03052* (2024).

[6] Amazon Web Services, Inc. *Amazon Web Services IoT TwinMaker*. en-US.

[7] A. G. Anastasiadis, A. Lekidis, A. Polyzakis, G. Vokas, and K. Hrissagis. “Stochastic multiobjective, non-convex CHPED, incorporating wind power and V2G under environmental constraints”. In: *Energy Systems* 16.4 (Nov. 2025), pp. 1263–1283. DOI: 10.1007/s12667-023-00611-1.

[8] Jason Ansel et al. “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation”. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS ’24)*. ACM, Apr. 2024. DOI: 10.1145/3620665.3640366.

[9] Donovan Artz and Yolanda Gil. “A survey of trust in computer science and the Semantic Web”. In: *Journal of Web Semantics* 5.2 (2007). Software Engineering and the Semantic Web, pp. 58–71. DOI: https://doi.org/10.1016/j.websem.2007.03.002.

[10] Rasool Azimi and Hedieh Sajedi. “A decentralized gossip based approach for data clustering in peer-to-peer networks”. In: *Journal of Parallel and Distributed Computing* 119 (2018), pp. 64–80. DOI: https://doi.org/10.1016/j.jpdc.2018.03.009.

[11] Radhakisan Baheti and Helen Gill. “Cyber-physical systems”. In: *The impact of control technology* 12.1 (2011), pp. 161–166.

[12] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”. In: *science* 286.5439 (1999), pp. 509–512.

[13] Dominique Barth, Benjamin Cohen-Boulakia, and Wilfried Ehounou. “Distributed Reinforcement Learning for the Management of a Smart Grid Interconnecting Independent Prosumers”. In: *Energies* 15.4 (2022). DOI: 10.3390/en15041440.

[14] Ammar Ayman Battah, Youssef Iraqi, and Ernesto Damiani. “A Trust and Reputation System for IoT Service Interactions”. In: *IEEE Transactions on Network and Service Management* 19.3 (2022), pp. 2987–3005. DOI: 10.1109/TNSM.2022.3179875.

[15] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. “The complexity of decentralized control of Markov decision processes”. In: *Mathematics of operations research* 27.4 (2002), pp. 819–840.

[16] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. “The Role of Trust Management in Distributed Systems Security”. In: *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*. Ed. by Jan Vitek and Christian D. Jensen. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 185–210. DOI: 10.1007/3-540-48749-2_8.

[17] Cara Bloom, Joshua Tan, Javed Ramjohn, and Lujo Bauer. “Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles”. In: *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. Santa Clara, CA: USENIX Association, July 2017, pp. 357–375.

[18] Stefan Boschert and Roland Rosen. “Digital twin - the simulation aspect”. In: *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*. Springer, 2016, pp. 59–74.

[19] Lucian Busoniu, Robert Babuska, and Bart De Schutter. “A comprehensive survey of multiagent reinforcement learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), pp. 156–172.

[20] Lal Verda Cakir, Craig J. Thomson, Mehmet Özdem, Berk Canberk, Van-Linh Nguyen, and Trung Q. Duong. “Intelligent Digital Twin Communication Framework for Addressing Accuracy and Timeliness Tradeoff in Resource-Constrained Networks”. In: *IEEE Transactions on Cognitive Communications and Networking* 11.3 (2025), pp. 1954–1965. DOI: 10.1109/TCCN.2024.3469234.

[21] Christiano Castelfranchi and Rino Falcone. *Trust theory: A socio-cognitive and computational model*. John Wiley & Sons, 2010.

[22] Yee Ting Chai, Hang Seng Che, ChiaKwang Tan, Wooi-Nee Tan, Sook-Chin Yip, and Ming-Tao Gan. “A two-stage optimization method for Vehicle to Grid coordination considering building and Electric Vehicle user expectations”. In: *International Journal of Electrical Power & Energy Systems* 148 (2023), p. 108984. DOI: 10.1016/j.ijepes.2023.108984.

[23] Kemboi Cheruiyot, Nickson Kiprotich, Vyacheslav Kungurtsev, Kennedy Mugo, Vivian Mwirigi, and Marvin Ngesa. *A Survey of Multi Agent Reinforcement Learning: Federated Learning and Cooperative and Noncooperative Decentralized Regimes*. 2025.

[24] Alessandro Costantini et al. “IoTwins: Toward Implementation of Distributed Digital Twins in Industry 4.0 Settings”. In: *Computers* 11.5 (2022). DOI: 10.3390/computers11050067.

[25] F. Darema, E. Blasch, S. Ravela, and A. Aved. *Dynamic Data Driven Applications Systems: Third International Conference, DDDAS 2020, Boston, MA, USA, October 2-4, 2020, Proceedings*. Lecture Notes in Computer Science. Springer International Publishing, 2020.

[26] Daniel Dewey. “Reinforcement Learning and the Reward Engineering Principle.” In: *AAAI spring symposia*. 2014.

[27] Ali Dorri, Salil S Kanhere, and Raja Jurdak. “Multi-agent systems: A survey”. In: *Ieee Access* 6 (2018), pp. 28573–28593.

[28] Wei Du and Shifei Ding. “A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications”. In: *Artificial Intelligence Review* 54.5 (2021), pp. 3215–3238.

[29] Rudresh Dwivedi et al. “Explainable AI (XAI): Core ideas, techniques, and solutions”. In: *ACM computing surveys* 55.9 (2023), pp. 1–33.

[30] Abir El Azzaoui, Sekione Reward Jeremiah, Neal N. Xiong, and Jong Hyuk Park. “A digital twin-based edge intelligence framework for decentralized decision in IoV system”. In: *Information Sciences* 649 (2023), p. 119595. DOI: https://doi.org/10.1016/j.ins.2023.119595.

[31] Xiaohan Fang, Qiang Zhao, Jinkuan Wang, Yinghua Han, and Yuchun Li. “Multi-agent Deep Reinforcement Learning for Distributed Energy Management and Strategy Optimization of Microgrid Market”. In: *Sustainable Cities and Society* 74 (2021), p. 103163. DOI: https://doi.org/10.1016/j.scs.2021.103163.

[32] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. “Learning to Communicate with Deep Multi-Agent Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016.

[33] Marisol García-Valls and Alejandro M Chirivella-Ciruelos. “CoTwin: Collaborative improvement of digital twins enabled by blockchain”. In: *Future Generation Computer Systems* (2024).

[34] Andrea Giulianelli, Samuele Burattini, Andrei Ciortea, and Alessandro Ricci. “HWoDT Framework: A toolchain to build interoperable Digital Twin Ecosystems”. In: *SoftwareX* 31 (2025), p. 102275. DOI: https://doi.org/10.1016/j.softx.2025.102275.

[35] Parham Gohari, Matthew Hale, and Ufuk Topcu. “Privacy-Engineered Value Decomposition Networks for Cooperative Multi-Agent Reinforcement Learning”. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023, pp. 8038–8044. DOI: 10.1109/CDC49753.2023.10384184.

[36] Michael Grieves. “Digital model, digital shadow, digital twin”. In: *Preprint* (2023).

[37] Michael Grieves and John Vickers. “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”. In: *Transdisciplinary perspectives on complex systems: New findings and approaches* (2017), pp. 85–113.

[38] Nils Gruschka, Vasileios Mavroeidis, Kamer Vishi, and Meiko Jensen. “Privacy Issues and Data Protection in Big Data: A Case Study Analysis under GDPR”. In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 5027–5033. DOI: 10.1109/BigData.2018.8622621.

[39] Félix Gómez Mármol and Gregorio Martínez Pérez. “Trust and reputation models comparison”. In: *Internet Research* 21.2 (Jan. 2011), pp. 138–153. DOI: 10.1108/10662241111123739.

[40] Omar Hasan. *A Survey of Privacy Preserving Reputation Systems*. Technical Report. LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, Nov. 2017.

[41] Ezz El-Din Hemdan and Amged Sayed. “Smart and Secure Healthcare with Digital Twins: A Deep Dive into Blockchain, Federated Learning, and Future Innovations”. In: *Algorithms* 18.7 (2025). DOI: 10.3390/a18070401.

[42] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. “A survey of attack and defense techniques for reputation systems”. In: *ACM Comput. Surv.* 42.1 (Dec. 2009). DOI: 10.1145/1592451.1592452.

[43] Daner Hu, Zhenhui Ye, Yuanqi Gao, Zuzhao Ye, Yonggang Peng, and Nanpeng Yu. “Multi-Agent Deep Reinforcement Learning for Voltage Control With Coordinated Active and Reactive Power Optimization”. In: *IEEE Transactions on Smart Grid* 13.6 (2022), pp. 4873–4886. DOI: 10.1109/TSG.2022.3185975.

[44] Zhengchang Hua, Karim Djemame, Nikos Tziritas, and Georgios Theodoropoulos. “A Framework for Digital Twin Collaboration”. In: *2024 Winter Simulation Conference (WSC)*. 2024.

[45] Zhengchang Hua, Panagiotis Oikonomou, Karim Djemame, Nikos Tziritas, and Georgios Theodoropoulos. “A Digital Twin-based Multi-Agent Reinforcement Learning Framework for Vehicle-to-Grid Coordination”. In: *The 25th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2025)*. 2025.

[46] Patrik Hummel, Matthias Braun, Max Tretter, and Peter Dabrock. “Data sovereignty: A review”. In: *Big Data & Society* 8.1 (2021), p. 2053951720982012.

[47] Muhammad Akbar Husnoo, Adnan Anwar, Md Enamul Haque, and A. N. Mahmood. *Decentralized Federated Anomaly Detection in Smart Grids: A P2P Gossip Approach*. 2025.

[48] Sergio Infante, Julia Robles, Cristian Martín, Bartolomé Rubio, and Manuel Díaz. "Distributed digital twins on the open-source OpenTwins framework". In: *Advanced Engineering Informatics* 64 (2025), p. 102970. DOI: https://doi.org/10.1016/j.aei.2024.102970.

[49] Nasser Jazdi. "Cyber physical systems in the context of Industry 4.0". In: *2014 IEEE international conference on automation, quality and testing, robotics*. IEEE. 2014, pp. 1–4.

[50] Christian Damsgaard Jensen. "The Importance of Trust in Computer Security". In: *Trust Management VIII*. Ed. by Jianying Zhou, Nurit Gal-Oz, Jie Zhang, and Ehud Gudes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–12.

[51] Jiechuan Jiang and Zongqing Lu. "Learning fairness in multi-agent systems". In: *Advances in Neural Information Processing Systems* 32 (2019).

[52] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. "Characterising the Digital Twin: A systematic literature review". In: *CIRP journal of manufacturing science and technology* 29 (2020), pp. 36–52.

[53] Audun Josang and Roslan Ismail. "The beta reputation system". In: *Proceedings of the 15th bled electronic commerce conference*. Vol. 5. 2002, pp. 2502–2511.

[54] Audun Jøsang, Roslan Ismail, and Colin Boyd. "A survey of trust and reputation systems for online service provision". In: *Decision support systems* 43.2 (2007), pp. 618–644.

[55] Peizhong Ju, Arnob Ghosh, and Ness Shroff. "Achieving Fairness in Multi-Agent MDP Using Reinforcement Learning". In: *The Twelfth International Conference on Learning Representations*. 2024.

[56] Ersan Kabalci and Yasin Kabalci. “Introduction to Smart Grid Architecture”. In: *Smart Grids and Their Communication Systems*. Ed. by Ersan Kabalci and Yasin Kabalci. Singapore: Springer Singapore, 2019, pp. 3–45. DOI: 10.1007/978-981-13-1768-2_1.

[57] Peter Kairouz et al. “Advances and open problems in federated learning”. In: *Foundations and trends® in machine learning* 14.1–2 (2021), pp. 1–210.

[58] Md. Kamruzzaman, Jiajun Duan, Di Shi, and Mohammed Benidris. “A Deep Reinforcement Learning-Based Multi-Agent Framework to Enhance Power System Resilience Using Shunt Resources”. In: *IEEE Transactions on Power Systems* 36.6 (2021), pp. 5525–5536. DOI: 10.1109/TPWRS.2021.3078446.

[59] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. “The eigentrust algorithm for reputation management in p2p networks”. In: *Proceedings of the 12th international conference on World Wide Web*. 2003, pp. 640–651.

[60] Ioannis Karakikes, Eftihia Nathanail, and Mihails Savrasovs. “Techniques for Smart Urban Logistics Solutions’ Simulation: A Systematic Review”. In: *Reliability and Statistics in Transportation and Communication*. Ed. by Igor Kabashkin, Irina Yatskiv (Jackiva), and Olegas Prentkovskis. Cham: Springer International Publishing, 2019, pp. 551–561.

[61] Charles B Keating, Polinpapilinho F Katina, and Joseph M Bradley. “Complex system governance: concept, challenges, and emerging research”. In: *International Journal of System of Systems Engineering* 5.3 (2014), pp. 263–288.

[62] D. Kempe, A. Dobra, and J. Gehrke. “Gossip-based computation of aggregate information”. In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* 2003, pp. 482–491. DOI: 10.1109/SFCS.2003.1238221.

[63] Latif U. Khan, Ehzaz Mustafa, Junaid Shuja, Faisal Rehman, Kashif Bilal, Zhu Han, and Choong Seon Hong. “Federated Learning for Digital Twin-Based Vehicular Networks: Architecture and Challenges”. In: *IEEE Wireless Communications* 31.2 (2024), pp. 156–162. DOI: 10.1109/MWC.012.2200373.

[64] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014).

[65] Vijay Konda and John Tsitsiklis. “Actor-critic algorithms”. In: *Advances in neural information processing systems* 12 (1999).

[66] Ricardo Eiji Kondo, Willian Jeferson Andrade, Clayton de Mello Henequim, André Eugenio Lazzaretti, Alceu de Souza Britto, Eduardo de Freitas Rocha Loures, Eduardo Alves Portela Santos, and Gilberto Reynoso-Meza. “An industrial edge computing architecture for Local Digital Twin”. In: *Computers & Industrial Engineering* 193 (2024), p. 110257. DOI: https://doi.org/10.1016/j.cie.2024.110257.

[67] Marija Kutelega, Renata Mekovec, and Ahmed Shareef. “Privacy and security challenges of the digital twin: systematic literature review”. In: *JUCS - Journal of Universal Computer Science* 30.13 (2024). Publisher: Journal of Universal Computer Science _eprint: https://doi.org/10.3897/jucs pp. 1782–1806. DOI: 10.3897/jucs.114607.

[68] Hyo-Cheol Lee and Seok-Won Lee. “Trust as Soft Security for Self-Adaptive Systems: A Literature Survey”. In: *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 2. 2017, pp. 523–528. DOI: 10.1109/COMPSAC.2017.255.

[69] Dun Li, Dezhi Han, Noel Crespi, Roberto Minerva, Syed Mohsan Raza, Reza Farahbakhsh, Wei Liang, and Zibin Zheng. “Blockchain in the Digital Twin Context: A Comprehensive Survey”. In: *ACM Comput. Surv.* 58.6 (Dec. 2025). DOI: 10.1145/3772366.

[70] Guanjie Li, Tom H. Luan, Xinghao Li, Jinkai Zheng, Chengzhe Lai, Zhou Su, and Kuan Zhang. “Breaking Down Data Sharing Barrier of Smart City: A Digital Twin Approach”. In: *IEEE Network* 38.1 (2024), pp. 238–246. DOI: 10.1109/MNET.140.2200512.

[71] Ming Li, Yelin Fu, Qiqi Chen, and Ting Qu. “Blockchain-enabled digital twin collaboration platform for heterogeneous socialized manufacturing resource management”. In: *International Journal of Production Research* 61.12 (2023), pp. 3963–3983. DOI: 10.1080/00207543.2021.1966118.

[72] Ming Li, Zhi Li, Xidian Huang, and Ting Qu. “Blockchain-based digital twin sharing platform for reconfigurable socialized manufacturing resource integration”. In: *International Journal of Production Economics* 240 (2021), p. 108223. DOI: https://doi.org/10.1016/j.ijpe.2021.108223.

[73] Yuanzheng Li, Chaofan Yu, Mohammad Shahidehpour, Tao Yang, Zhigang Zeng, and Tianyou Chai. “Deep Reinforcement Learning for Smart Grid Operations: Algorithms, Applications, and Prospects”. In: *Proceedings of the IEEE* 111.9 (2023), pp. 1055–1096. DOI: 10.1109/JPROC.2023.3303358.

[74] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. *Continuous control with deep reinforcement learning*. 2019.

[75] Yan Lin, Jinming Bao, Yijin Zhang, Jun Li, Feng Shu, and Lajos Hanzo. “Privacy-Preserving Joint Edge Association and Power Optimization for the Internet of Vehicles via Federated Multi-Agent Reinforcement Learning”. In: *IEEE Transactions on Vehicular Technology* 72.6 (2023), pp. 8256–8261. DOI: 10.1109/TVT.2023.3240682.

[76] Chunhua Liu, KT Chau, Diyun Wu, and Shuang Gao. “Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies”. In: *Proceedings of the IEEE* 101.11 (2013), pp. 2409–2427.

[77] Ruhan Liu, Tom H. Luan, Youyang Qu, Yong Xiang, Longxiang Gao, and Dongmei Zhao. “Internet of Digital Twin: Framework, Applications, and Enabling Technologies”. In: *IEEE Communications Surveys & Tutorials* 28 (2026), pp. 3870–3910. DOI: 10.1109/COMST.2025.3554579.

[78] Siamak Lotfi, Mostafa Sedighizadeh, Rezvan Abbasi, and Seyed Hossein Hosseinian. “Vehicle-to-grid bidding for regulation and spinning reserve markets: A robust optimal coordinated charging approach”. In: *Energy Reports* 11 (2024), pp. 925–936. DOI: 10.1016/j.egyr.2023.12.044.

[79] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”. In: *Neural Information Processing Systems (NIPS)* (2017).

[80] Om Malik et al. “Smart Grid Origins, Definitions, Technologies, and Emerging Trends: A Power Community Perspective”. In: *Journal of Modern Power Systems and Clean Energy* 13.6 (2025), pp. 1845–1853. DOI: 10.35833/MPCE.2025.000807.

[81] Asimina Marousi and Vassilis M. Charitopoulos. “Game theoretic optimisation in process and energy systems engineering: A review”. In: *Frontiers in Chemical Engineering* Volume 5 - 2023 (2023). DOI: 10.3389/fceng.2023.1130568.

[82] Maggie Mashaly. “Connecting the twins: A review on digital twin technology & its networking requirements”. In: *Procedia Computer Science* 184 (2021), pp. 299–305.

[83] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Arcas, and Aguera. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[84] Albert Meijer and Manuel Pedro Rodríguez Bolívar. "Governing the smart city: a review of the literature on smart urban governance". In: *International review of administrative sciences* 82.2 (2016), pp. 392–408.

[85] Chun Ding Meimin Wang Zhili Zhou. "Blockchain-Based Decentralized Reputation Management System for Internet of Everything in 6G-Enabled Cybertwin Architecture". In: *Journal of New Media* 3.4 (2021), pp. 137–150. DOI: 10.32604/jnm.2021.024543.

[86] Microsoft Corporation. *Azure Digital Twins*. en-US.

[87] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[88] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 3rd ed. 2025.

[89] Mohammad Momani and Subhash Challa. *Survey of trust models in different network domains*. 2010.

[90] Mandana Moshrefzadeh, Thomas Machl, David Gackstetter, Andreas Donaubauer, and Thomas H Kolbe. "Towards a distributed digital twin of the agricultural landscape". In: *Journal of Digital Landscape Architecture* 5 (2020), pp. 173–118.

[91] Ananta Mukherjee, Peeyush Kumar, Boling Yang, Nishanth Chandran, and Divya Gupta. *Privacy Preserving Multi-Agent Reinforcement Learning in Supply Chains*. 2023.

[92] Divya Nagaraj, Priya Khandelwal, Sandra Steyaert, and Olivier Gevaert. “Augmenting digital twins with federated learning in medicine”. In: *The Lancet Digital Health* 5.5 (May 2023). Publisher: Elsevier, e251–e253. DOI: 10.1016/S2589-7500(23)00044-4.

[93] Stefan Neumeier, Philipp Wintersberger, Anna-Katharina Frison, Armin Becher, Christian Facchi, and Andreas Riener. “Teleoperation: The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters”. In: *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI ’19. Utrecht, Netherlands: Association for Computing Machinery, 2019, pp. 186–197. DOI: 10.1145/3342197.3344534.

[94] Hoang N. T. Nguyen, Cishen Zhang, and Md. Apel Mahmud. “Optimal Coordination of G2V and V2G to Support Power Grids With High Penetration of Renewable Energy”. In: *IEEE Transactions on Transportation Electrification* 1.2 (2015), pp. 188–195. DOI: 10.1109/TTE.2015.2430288.

[95] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer, 2016.

[96] V. Padmavathi, R. Kanimozhi, and R. Saminathan. “Digital twin driven smart factories: real time physics based co-simulation using edge a.i. and federated learning”. In: *Scientific Reports* 15.1 (Dec. 2025), p. 43373. DOI: 10.1038/s41598-025-28466-9.

[97] Junjie Pang, Yan Huang, Zhenzhen Xie, Jianbo Li, and Zhipeng Cai. “Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution”. In: *Tsinghua Science and Technology* 26.5 (2021), pp. 759–771. DOI: 10.26599/TST.2021.9010026.

[98] Keonwoo Park and Ilkyeong Moon. “Multi-agent deep reinforcement learning approach for EV charging scheduling in a smart grid”. In: *Applied Energy* 328 (2022), p. 120111. DOI: https://doi.org/10.1016/j.apenergy.2022.120111.

[99] Scott B Peterson, Jay Apt, and JF Whitacre. “Lithium-ion battery cell degradation resulting from realistic vehicle and vehicle-to-grid utilization”. In: *Journal of Power Sources* 195.8 (2010), pp. 2385–2392.

[100] Marco Picone, Marco Mamei, and Franco Zambonelli. “A Flexible and Modular Architecture for Edge Digital Twin: Implementation and Evaluation”. In: *ACM Trans. Internet Things* 4.1 (Feb. 2023). DOI: 10.1145/3573206.

[101] Caroline Player and Nathan Griffiths. “Improving trust and reputation assessment with dynamic behaviour”. In: *The Knowledge Engineering Review* 35 (2020), e29. DOI: 10.1017/S0269888920000077.

[102] Danny Pudjianto, Charlotte Ramsay, and Goran Strbac. “Virtual power plant and system integration of distributed energy resources”. In: *IET Renewable power generation* 1.1 (2007), pp. 10–16.

[103] Benedikt Putz, Marietheres Dietz, Philip Empl, and Günther Pernul. “EtherTwin: Blockchain-based Secure Digital Twin Information Management”. In: *Information Processing & Management* 58.1 (2021), p. 102425. DOI: https://doi.org/10.1016/j.ipm.2020.102425.

[104] Swarna Priya Ramu, Parimala Boopalan, Quoc-Viet Pham, Praveen Kumar Reddy Maddikunta, Thien Huynh-The, Mamoun Alazab, Thanh Thi Nguyen, and Thippa Reddy Gadekallu. “Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions”. In: *Sustainable Cities and Society* 79 (2022), p. 103663. DOI: https://doi.org/10.1016/j.scs.2021.103663.

[105] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*. 2018.

[106] Magnus Redeker, Jan Nicolas Weskamp, Bastian Rössl, and Florian Pethig. "Towards a Digital Twin Platform for Industrie 4.0". In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021, pp. 39–46. DOI: 10.1109/ICPS49255.2021.9468204.

[107] Alessandro Ricci, Angelo Croatti, Stefano Mariani, Sara Montagna, and Marco Picone. "Web of Digital Twins". In: *ACM Trans. Internet Technol.* 22.4 (Nov. 2022). DOI: 10.1145/3507909.

[108] Martin Roesch, Christian Linder, Roland Zimmermann, Andreas Rudolf, Andrea Hohmann, and Gunther Reinhart. "Smart Grid for Industry Using Multi-Agent Reinforcement Learning". In: *Applied Sciences* 10.19 (2020). DOI: 10.3390/app10196900.

[109] Radhya Sahal, Saeed H Alsamhi, Kenneth N Brown, Donna O'Shea, and Bader Alouffi. "Blockchain-based digital twins collaboration for smart pandemic alerting: decentralized COVID-19 pandemic alerting use case". In: *Computational Intelligence and Neuroscience* 2022 (2022).

[110] Akshita Maradapu Vera Venkata Sai, Chenyu Wang, Zhipeng Cai, and Yingshu Li. "Navigating the Digital Twin Network landscape: A survey on architecture, applications, privacy and security". In: *High-Confidence Computing* 4.4 (2024), p. 100269. DOI: https://doi.org/10.1016/j.hcc.2024.100269.

[111] Omer San, Suraj Pawar, and Adil Rasheed. "Decentralized digital twins of complex dynamical systems". In: *Scientific Reports* 13.1 (2023), p. 20087.

[112] Benjamin K Sovacool, Jonn Axsen, and Willett Kempton. “The future promise of vehicle-to-grid (V2G) integration: a sociotechnical review and research agenda”. In: *Annual Review of Environment and Resources* 42.1 (2017), pp. 377–406.

[113] Wen Sun, Shiyu Lei, Lu Wang, Zhiqiang Liu, and Yan Zhang. “Adaptive Federated Learning and Digital Twin for Industrial Internet of Things”. In: *IEEE Transactions on Industrial Informatics* 17.8 (2021), pp. 5605–5614. DOI: 10.1109/TII.2020.3034674.

[114] Wen Sun, Ning Xu, Lu Wang, Haibin Zhang, and Yan Zhang. “Dynamic Digital Twin and Federated Learning With Incentives for Air-Ground Networks”. In: *IEEE Transactions on Network Science and Engineering* 9.1 (2022), pp. 321–333. DOI: 10.1109/TNSE.2020.3048137.

[115] Peter Sunehag et al. “Value-decomposition networks for cooperative multi-agent learning”. In: *arXiv preprint arXiv:1706.05296* (2017).

[116] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.

[117] Gayatri Swamynathan, Kevin C. Almeroth, and Ben Y. Zhao. “The design of a reliable reputation system”. In: *Electronic Commerce Research* 10.3 (Dec. 2010), pp. 239–270. DOI: 10.1007/s10660-010-9064-y.

[118] Kang Miao Tan, Vigna K Ramachandaramurthy, and Jia Ying Yong. “Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques”. In: *Renewable and Sustainable Energy Reviews* 53 (2016), pp. 720–732.

[119] Kang Miao Tan, Vigna K Ramachandaramurthy, and Jia Ying Yong. “Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques”. In: *Renewable and Sustainable Energy Reviews* 53 (2016), pp. 720–732.

[120] Fengxiao Tang, Xuehan Chen, Tiago Koketsu Rodrigues, Ming Zhao, and Nei Kato. “Survey on Digital Twin Edge Networks (DITEN) Toward 6G”. In: *IEEE Open Journal of the Communications Society* 3 (2022), pp. 1360–1381. DOI: 10.1109/OJCOMS.2022.3197811.

[121] Fei Tao, Qinglin Qi, Lihui Wang, and AYC Nee. “Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison”. In: *Engineering* 5.4 (2019), pp. 653–661.

[122] Fei Tao, Yongping Zhang, Ying Cheng, Jiawei Ren, Dongxu Wang, Qinglin Qi, and Pei Li. “Digital twin and blockchain enhanced smart manufacturing service collaboration and management”. In: *Journal of Manufacturing Systems* 62 (2022), pp. 903–914. DOI: https://doi.org/10.1016/j.jmsy.2020.11.008.

[123] Mark Towers et al. “Gymnasium: A Standard Interface for Reinforcement Learning Environments”. In: *arXiv preprint arXiv:2407.17032* (2024).

[124] Ikram Ud Din, Aniqa Bano, Kamran Ahmad Awan, Ahmad Almogren, Ayman Altameem, and Mohsen Guizani. “LightTrust: Lightweight Trust Management for Edge Devices in Industrial Internet of Things”. In: *IEEE Internet of Things Journal* 10.4 (2023), pp. 2776–2783. DOI: 10.1109/JIOT.2021.3081422.

[125] Thomas H-J Uhlemann, Christian Lehmann, and Rolf Steinhilper. “The digital twin: Realizing the cyber-physical production system for industry 4.0”. In: *Procedia Cirp* 61 (2017), pp. 335–340.

[126] Faizan Ullah, Abdu Salam, Farhan Amin, Izaz Ahmad Khan, Jamal Ahmed, Shamzash Alam Zaib, and Gyu Sang Choi. “Deep Trust: A Novel Framework for Dynamic Trust and Reputation Management in the Internet of Things (IoT)-Based Networks”. In: *IEEE Access* 12 (2024), pp. 87407–87419. DOI: 10.1109/ACCESS.2024.3409273.

[127] Koen Van Heuveln, Rishabh Ghotge, Jan Anne Annema, Esther van Bergen, Bert van Wee, and Udo Pesch. “Factors influencing consumer acceptance of vehicle-to-grid by electric vehicle drivers in the Netherlands”. In: *Travel Behaviour and Society* 24 (2021), pp. 34–45.

[128] Eric VanDerHorn and Sankaran Mahadevan. “Digital Twin: Generalization, characterization and implementation”. In: *Decision support systems* 145 (2021), p. 113524.

[129] Praneeth Vepakomma. *Connecting silos with distributed and private computation*. Massachusetts Institute of Technology, 2024.

[130] Christian Vergara, Rami Bahsoon, Georgios Theodoropoulos, Wendy Yanez, and Nikos Tziritas. “Federated Digital Twin”. In: *2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. 2023, pp. 115–116. DOI: 10.1109/DS-RT58998.2023.00027.

[131] Christian Roberto Vergara, Georgios Theodoropoulos, Rami Bahsoon, Wendy Yanez, and Nikos Tziritas. “Federated Digital Twins as an Enabling Technology for Collaborative Decision-Making”. In: *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. SIGSIM-PADS ’24. Atlanta, GA, USA: Association for Computing Machinery, 2024, pp. 6768. DOI: 10.1145/3615979.3662152.

[132] Christian Vergara-Marcillo, Rami Bahsoon, Nikos Tziritas, and Georgios Theodoropoulos. “A Connectionist Approach to Federated Digital Twins”. In: *Computational Science – ICCS 2025*. Cham: Springer Nature Switzerland, 2025, pp. 60–74.

[133] A Villalonga, E Negri, L Fumagalli, M Macchi, F Castaño, and R Haber. “Local decision making based on distributed digital twin framework”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 10568–10573.

[134] Mitchell M Waldrop. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, 1993.

[135] Chenhao Wang, Yang Ming, Hang Liu, Jie Feng, and Ning Zhang. “Secure and Flexible Data Sharing With Dual Privacy Protection in Vehicular Digital Twin Networks”. In: *IEEE Transactions on Intelligent Transportation Systems* 25.9 (2024), pp. 12407–12420. DOI: 10.1109/TITS.2024.3368342.

[136] Yuntao Wang, Zhou Su, Shaolong Guo, Minghui Dai, Tom H Luan, and Yiliang Liu. “A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects”. In: *IEEE Internet of Things Journal* 10.17 (2023), pp. 14965–14987.

[137] Paul J. Werbos. “Computational Intelligence for the Smart Grid-History, Challenges, and Opportunities”. In: *IEEE Computational Intelligence Magazine* 6.3 (2011), pp. 14–21. DOI: 10.1109/MCI.2011.941587.

[138] Yiwen Wu, Ke Zhang, and Yan Zhang. “Digital twin networks: A survey”. In: *IEEE Internet of Things Journal* 8.18 (2021), pp. 13789–13804.

[139] Li Xiong and Ling Liu. “PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities”. In: *IEEE Transactions on Knowledge and Data Engineering* 16.7 (2004), pp. 843–857. DOI: 10.1109/TKDE.2004.1318566.

[140] Ziming Yan and Yan Xu. "A Multi-Agent Deep Reinforcement Learning Method for Cooperative Load Frequency Control of a Multi-Area Power System". In: *IEEE Transactions on Power Systems* 35.6 (2020), pp. 4599–4608. DOI: 10.1109/TPWRS.2020.2999890.

[141] Helin Yang, Jun Zhao, Zehui Xiong, Kwok-Yan Lam, Sumei Sun, and Liang Xiao. "Privacy-Preserving Federated Learning for UAV-Enabled Networks: Learning-Based Joint Scheduling and Resource Management". In: *IEEE Journal on Selected Areas in Communications* 39.10 (2021), pp. 3144–3159. DOI: 10.1109/JSAC.2021.3088655.

[142] Yujian Ye, Yi Tang, Huiyu Wang, Xiao-Ping Zhang, and Goran Strbac. "A Scalable Privacy-Preserving Multi-Agent Deep Reinforcement Learning Approach for Large-Scale Peer-to-Peer Transactive Energy Trading". In: *IEEE Transactions on Smart Grid* 12.6 (2021), pp. 5185–5200. DOI: 10.1109/TSG.2021.3103917.

[143] Seungwook Yoon, Kanggu Park, and Euiseok Hwang. "Connected electric vehicles for flexible vehicle-to-grid (V2G) services". In: *2017 International Conference on Information Networking (ICOIN)*. 2017, pp. 411–413. DOI: 10.1109/ICOIN.2017.7899469.

[144] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*. 2022.

[145] Tao Yu, Zongdian Li, Omar Hashash, Kei Sakaguchi, Walid Saad, and Mérouane Debbah. "Internet of Federated Digital Twins: Connecting Twins Beyond Borders for Society 5.0". In: *IEEE Internet of Things Magazine* 7.5 (2024), pp. 64–71. DOI: 10.1109/IOTM.001.2300279.

[146] Tingting Yuan, Hwei-Ming Chung, and Xiaoming Fu. *PP-MARL: Efficient Privacy-Preserving Multi-Agent Reinforcement Learning for Cooperative Intelligence in Communications*. 2025.

[147] Zerotier. *A smart ethernet switch for Earth*. 2024.

[148] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. "A survey on federated learning". In: *Knowledge-Based Systems* 216 (2021), p. 106775.

[149] Nan Zhang, Rami Bahsoon, and Georgios Theodoropoulos. "Towards Engineering Cognitive Digital Twins with Self-Awareness". In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2020, pp. 3891–3891. DOI: 10.1109/SMC42975.2020.9283357.

[150] Nan Zhang, Rami Bahsoon, Nikos Tziritas, and Georgios Theodoropoulos. "Knowledge Equivalence in Digital Twins of Intelligent Systems". In: *ACM Trans. Model. Comput. Simul.* 34.1 (2024). DOI: 10.1145/3635306.

[151] Yaoyu Zhang, Jian Sun, and Chenye Wu. "Vehicle-to-Grid Coordination via Mean Field Game". In: *IEEE Control Systems Letters* 6 (2022), pp. 2084–2089. DOI: 10.1109/LCSYS.2021.3139266.

[152] Canzhe Zhao, Yanjie Ze, Jing Dong, Baoxiang Wang, and Shuai Li. *DPMAC: Differentially Private Communication for Cooperative Multi-Agent Reinforcement Learning*. 2023.

[153] Zhe Zhou, Xue Li, Huaichang Ge, Jiahui Zhang, and Yixun Xue. "Congestion-Aware Rebalancing and Vehicle-to-Grid Coordination of Shared Electric Vehicles: An Aggregative Game Approach". In: *IEEE Transactions on Transportation Electrification* 11.1 (2025), pp. 275–285. DOI: 10.1109/TTE.2024.3389712.

[154] Ligeng Zhu, Zhijian Liu, and Song Han. "Deep leakage from gradients". In: *Advances in neural information processing systems* 32 (2019).