



ARTICLE

Quantitative Stress Testing Using Scalable Digital Twin Simulation with MobileX Pole for Intelligent Mobile Surveillance

DongHwan Ku, Sun Park* and JongWon Kim

AI Graduate School, GIST, Gwangju, Republic of Korea

*Corresponding Author: Sun Park. Email: sunpark@gist.ac.kr

Received: 23 January 2026; Accepted: 17 April 2026; Published: 15 June 2026

ABSTRACT: In future smart cities, ensuring urban safety requires data-driven decision-making through real-time monitoring tailored to dynamic, complex environments. Such surveillance relies on diverse mobile sensor devices, including drones, robots, patrol vehicles, and portable sensors. However, scaling and validating these systems directly in the real world is constrained by high costs, safety risks, and limited reproducibility across operating conditions. A scalable Digital Twin (DT) model can overcome these constraints by reproducing real-world mobile surveillance in a virtual environment, enabling large-scale simulations of sensor deployment, communication scenarios, and high-density visual data processing. Nevertheless, digital twins still face well-known limitations such as the reality gap, construction costs, limited coverage of behavioral and social variables, biased learning in AI models, and the need for continuous updates. Many of these issues are expected to be mitigated in the near future as generative AI increasingly automates the construction of virtual environments and objects. Despite these advancements, the systemic resource constraints of integrating large-scale physical sensor streams with virtual rendering remain underexplored. To address this gap, this paper proposes a scalable DT framework for the quantitative stress testing of intelligent mobile surveillance systems. The proposed framework collects real-world visualization data from multiple cameras mounted on MobileX Poles, and supports quantitative stress testing in both virtual and physical environments. It systematically analyzes how computing resource usage varies with the number of smart poles and the total number of camera streams under rendering conditions, thereby quantifying the resource limits of real-world, multi-camera DT simulations.

KEYWORDS: Quantitative stress testing; scalable digital twin; MobileX Pole; intelligent mobile surveillance

1 Introduction

With the proliferation of smart cities, urban management has evolved beyond reactive responses to proactive response systems based on data-driven decision making. Smart cities are increasingly reliant on real-time surveillance systems to ensure the safety, operational efficiency, and sustainability of complex and dynamic urban environments [1–3]. However, the fixed CCTV infrastructure commonly used in existing surveillance systems has structural limitations, such as blind spots and limited mobility. To overcome these limitations, mobile surveillance systems (MSS), such as autonomous drones, patrol robots, and mobile smart poles, have emerged as essential solutions for modern smart city management [4,5]. The authors of this paper previously developed a prototype of the “MobileX Pole”, a multi-purpose mobile sensing device, as part of this solution. This pole is designed to provide 360-degree situational awareness using six global shutter cameras [6–8]. However, these MSSs face significant challenges when scaled from the prototype stage to a city-scale.

Scaling up to smart cities is often impractical due to the prohibitive cost of large-scale field testing, safety concerns in densely populated areas, and a lack of reproducibility in real-world environments [9,10]. In addition, environmental factors such as lighting, pedestrian flow, and traffic patterns fluctuate unpredictably, making it difficult to conduct controlled experiments or reliably measure improvements in AI performance and communication protocols [11]. Digital twin (DT) technology offers an innovative way to overcome these limitations by creating high-fidelity virtual replicas of physical assets within a physics-based environment [12]. This facilitates risk-free system optimization by enabling large-scale virtual experiments, such as testing dense multi-camera network loads and diverse communication scenarios, which are difficult to perform consistently in the real-world. Despite these advantages, existing DT technologies have the following limitations: First, creating accurate digital replicas requires extensive manual modeling and continuous updates [13–15]. Second, the reality gap, which is the difference between simulated and real-world behaviors caused by stochastic factors, frequently causes the Sim2Real discrepancy, which prevents AI models from generalizing [16].

Recent advances in generative AI (GenAI), such as Neural Radiance Fields (NeRF) and generative 3D pipelines, are expected to overcome the challenges of DT by automatically generating realistic environments from real-world visual data [17,18]. Applying generative AI technology to DT for mobile surveillance systems can build a continuously updated virtual environment using various relevant data collected from a given environment [19,20]. However, these technologies can also create bottlenecks and computational overhead. For example, a DT environment for a mobile surveillance system comprised of numerous surveillance devices generating high-resolution video streams places a significant load on the simulation engine. Rendering and sensor emulation in visually complex, multi-agent environments, as shown in the example, consume significant infrastructure resources. Previous digital twin studies have primarily focused on functional demonstrations or static visual representations. They rarely address the systemic resource constraints, such as network bottlenecks and GPU VRAM saturation. These constraints emerge when integrating real-time physical video streams with large-scale virtual rendering. To bridge this gap, it is essential to thoroughly understand the computing resource consumption in both physical and virtual spaces.

This paper proposes a quantitative stress testing method for assessing the computing resource consumption of a digital twin-based mobile surveillance system (MSS) in both physical and virtual spaces. For quantitative testing of computing resources, this paper designs and implements a scalable digital twin framework based on MobileX Pole. The proposed framework and testing method facilitate the tracking of computing resources related to the performance and scalability of digital twins for mobile surveillance systems. The contributions of this paper are as follows: First, rather than proposing a standalone AI algorithm, we design a Kubernetes and Lakehouse-based scalable digital twin framework architecture. This provides an end-to-end evaluation pipeline that seamlessly integrates physical sensor streams with virtual environments based on a real MobileX Pole. Second, we propose a large-scale stress testing methodology that systematically explores the system resource limitations of future city-scale surveillance systems by expanding the number of virtual mobile poles to generate high-resolution video workloads. The proposed testing can track the correlation between network bandwidth and VRAM occupancy as a result of physical stream transmission and virtual simulation expansion.

The remainder of this paper is organized as follows: [Section 2](#) describes intelligent surveillance systems for smart cities, digital twin technology, and prior research on digital twin stress testing. [Section 3](#) describes the system architecture and the construction process of the scalable digital twin framework for quantitative stress testing. [Section 4](#) presents the experimental setup and scalability evaluation results. [Section 5](#) concludes with a discussion of key research findings and future research directions.

2 Related Works

Intelligent mobile surveillance systems in smart city environments are complex systems that must simultaneously address complex physical spatial structures, massive video data processing, and real-time analysis requirements. Due to these characteristics, recent research is expanding beyond simply focusing on improving the performance of individual surveillance algorithms to consider the integrated structure and stable operation of the entire system. In particular, surveillance systems that incorporate mobile surveillance platforms and city-scale infrastructure must simultaneously consider load variations occurring in real-world environments, limited resource conditions, and system scalability issues. The literature discussed in this section was primarily identified through thematic keyword searches focusing on “intelligent mobile surveillance”, “smart city digital twins”, and “digital twin stress testing”. Based on this scope, this section describes related approaches in three categories.

Intelligent Mobile Surveillance includes IoT-based video analytics [3,21], multi-camera tracking [22,23], mobile surveillance platforms [24–26], and system-level gap analyses [9,10]. Digital Twin for Smart City Surveillance covers foundational DT concepts, architectures, and research trends [13,27,28], smart-city DT surveys [29,30], city-scale decision-support frameworks [31–34], visualization and 3D representation techniques [35,36], high-load video workload limitations [14,15], and edge-AI-integrated DT approaches [37–39]. Stress Testing for Digital Twins comprises DT evaluation and simulation-centric architectures [40–42], and large-scale stress-testing methods covering HPC parallel simulation, scenario prioritization, and platform bottleneck analysis [43–45].

2.1 Intelligent Mobile Surveillance

Intelligent video surveillance technology has evolved into a core technology supporting public safety, traffic management, and anomaly detection in smart city environments. Current video detection research is expanding beyond traditional fixed camera surveillance to incorporate multi-camera environments, mobile surveillance platforms, and deep learning-based video analytics. Yanjinkham and Kim analyzed video surveillance infrastructure deployed across a city and the major service flows utilizing it from a smart city perspective [3]. This study comprehensively summarizes the overall technology of urban surveillance systems, including deep learning-based video analysis, edge-cloud connectivity, and data flow in IoT environments. Similar research [21] proposed a structure where large-scale camera streams are collected over a network and processed using distributed computational resources, highlighting the systematic nature of surveillance systems through IoT-based video analytics. Multi-object and multi-camera tracking technologies are key elements of intelligent surveillance systems, and recent research has focused on deep learning-based approaches. Fei and Han [22] and Amosa et al. [23] comprehensively analyzed continuous object tracking techniques in multi-camera environments, addressing issues such as feature representation, inter-camera alignment, and spatiotemporal linkage. These studies contributed to improving object tracking accuracy; discussions on city-level operation and system scalability are limited.

Additionally, research on mobile surveillance systems is being conducted to complement the limitations of fixed surveillance infrastructure. A UAV (Unmanned Aerial Vehicle) based intelligent mobile surveillance system offers a flexible structure for observing wide areas and demonstrates a case study combining aerial video-based crowd analysis and object recognition [24]. Furthermore, mobile robot-based surveillance systems offer the potential for autonomous surveillance in indoor and outdoor environments through scenarios that simultaneously consider obstacle avoidance and crowd surveillance [25]. There is research on intelligent surveillance support systems that integrate real-time threat detection and surveillance support functions [26]. However, this intelligent mobile surveillance research often focuses on individual surveillance

technologies or specific scenarios, which lack research on analyzing the entire surveillance system in a virtual environment or on operational load in large-scale urban environments [9,10].

2.2 Digital Twin for Smart City Surveillance

Digital twins, a technology that links physical systems with virtual models to enable real-time status reflection and simulation, have recently become a key research topic in the smart city field. Fuller et al. summarized the components and technical challenges of digital twins and proposed the concept of digital twins as a system integrating data collection, virtual modeling, analysis, and decision-making capabilities [27]. Liu et al. proposed a general structure for designing and operating digital twin systems by dividing digital twins into physical entities, virtual models, twin data, and application areas [28]. A large-scale quantitative study analyzing digital twin research trends in smart city environments [13] shows that digital twins are expanding into city-scale complex systems by combining IoT, artificial intelligence, big data, and simulation technologies.

Within this trend, numerous studies are utilizing smart city digital twins as a key tool to support urban infrastructure management and service operation. Huzzat et al. [29] and Yessef et al. [30] comprehensively summarized the technical components and application cases of smart city digital twins and analyzed their potential for use in various urban service areas, such as transportation, energy, environment, and public safety. Li et al. [31] conducted a study that further elaborated on the concept and structure of city-scale digital twins. This study presents the core concepts of bidirectional mapping and continuous data synchronization between physical and virtual cities and describes the basic architecture of a city-scale digital twin. Wang et al. [32] and Lv et al. [33] discussed ways to utilize digital twins as decision-making support tools for urban design and operation, emphasizing the integration of diverse urban data and services. A study [34] that built digital twins based on real-world urban cases and encouraged citizen participation demonstrated that digital twins can be utilized beyond simple visualization tools and serve as platforms to support policymaking and operational feedback. Recently, research on visualization and spatial representation technologies to enhance the realism and usability of digital twins has been increasing. Research combining sensor data and visual analytics [35] presented a method for visually representing real-time data streams in a digital twin environment. They also expanded the possibilities of virtual space-based analytics. Neural-based 3D representation techniques, such as NeRF [17] and 3D Gaussian Splatting [36], enable high-resolution virtual representations of urban environments. These technologies are attracting attention as technologies that can significantly improve the visual quality of smart city digital twins. Smart city digital twin research primarily focuses on urban infrastructure management, service simulation, or improving visualization quality. Therefore, the system-level load characteristics and operational stability associated with integrating high-load data sources, such as large-scale video surveillance systems, into digital twins are limited [13–15]. There is a lack of systematic research on how digital twin systems operate and what limitations they present, especially in situations where real-time video streams are flowing in at a large scale.

Recently, to address the scarcity of resources and the extreme need for efficiency, new architectural approaches integrating Edge AI and digital twins have emerged. For instance, Fan et al. [37] proposed a DT-empowered edge AI approach for vehicle control, Barbutto et al. [38] explored opportunistic DTs using Lingua Franca for distributed coordination, and Rahman et al. [39] reviewed machine learning applications with DT and edge AI integration. While these studies highlight efficient resource management in edge-DT ecosystems, they primarily focus on conceptual frameworks, algorithmic optimization, or isolated sensing. They still lack a quantitative analysis of system-level bottlenecks when handling continuous, high-load video streams from multiple mobile sensors.

2.3 Stress Testing for Digital Twins

For digital twin systems to be reliably utilized in real-world environments, stress testing is essential to analyze system behavior under various load conditions and operating scenarios. Tang et al. proposed a framework for evaluating domain-specific digital twin platforms, which compared and analyzed digital twin systems based on performance and reliability metrics [40]. This study emphasizes that digital twin evaluation should extend beyond simple functional verification to quantitative system analysis. Simulation-based techniques are a key approach for performing digital twin stress testing. Santos et al. proposed a simulation-centric digital twin architecture and analyzed system performance through various operating scenarios [41]. A similar study [42] generated realistic workloads in a large-scale simulation environment for a smart city software platform and evaluated system scalability and bottlenecks. These studies show that stress testing is a key element in digital twin system design.

Recently, research has been reported on analyzing extreme load conditions by performing large-scale digital twin simulations in high-performance computing environments. Samak et al. analyzed the computational load and scalability of a large-scale digital twin environment using parallel simulation [43]. Nasrazadani et al. [44] proposed a method for efficiently designing stress test scenarios, which prioritizes test combinations that effectively expose system vulnerabilities, rather than randomly testing all possible scenarios. Li et al. [45] conducted a study analyzing the performance and bottlenecks of the simulation environment itself. They demonstrated that digital twin-based experiments must consider not only the system being evaluated but also the limitations of the simulation platform. These studies imply that digital twin stress testing is not simply a functional experiment, but a comprehensive evaluation process that encompasses system-wide resource usage and scalability. However, previous stress testing research has been limited to analyzing data throughput in single domains such as networks, transportation, and energy [42–44], or has limited itself to measuring simulation loads based on static data. These studies still lack the ability to analyze system limitations in composite workload environments where high-resolution video decoding, real-time physics engine computation, and 3D rendering loads occur simultaneously. In intelligent mobile surveillance for smart cities, when large-scale visual data is projected from physical to virtual space in real-time, it is difficult to find examples that quantitatively identify bottlenecks throughout the entire pipeline, from data ingestion to visualization (i.e., rendering).

To summarize the research gaps identified across intelligent mobile surveillance, edge AI integrations, and digital twin stress testing, Table 1 compares our proposed framework with representative recent studies. Unlike existing works that focus primarily on conceptual architectures or simulated workloads, our approach uniquely integrates high-load, multi-camera physical streams with virtual environments to quantitatively evaluate the end-to-end scalability limits of DT systems.

Table 1: Comparison of scalable digital twin frameworks for quantitative stress testing.

Reference	Application Domain	Real-Time Data Integration	Multi-Stream Visual Workload	Quantitative Stress Testing	Main Contribution
Fan et al. [37]	Vehicular Networks	Partial (Synthetic data)	No	No	DT-empowered Edge AI resource allocation
Barbuto et al. [38]	Distributed Systems	No	No	No	Opportunistic DT coordination via Lingua Franca

(Continued)

Table 1 (continued)

Reference	Application Domain	Real-Time Data Integration	Multi-Stream Visual Workload	Quantitative Stress Testing	Main Contribution
Samak et al. [43]	Autonomous Driving	No	No	Yes (Static data)	Parallel simulation scalability in HPC
Li et al. [45]	Network Systems	No	No	Yes (Network only)	Performance bottleneck analysis of simulation platforms
Proposed Framework	Mobile Surveillance	Yes (Dual-space mapping)	Yes (High-fidelity rendering)	Yes (End-to-End pipeline)	Scalable DT framework for quantitative stress testing

3 Scalable Digital Twin Framework for Quantitative Stress Testing

This Section proposes the overall architecture of a scalable digital twin framework for quantitative stress testing of Intelligent Mobile Surveillance. This framework aims to overcome the spatial and temporal constraints of physical testing environments and enable quantitative performance evaluation by integrating real-world surveillance data collected in real-world environments with large-scale simulation data generated in virtual environments. Fig. 1 shows the overall architecture of the proposed framework. The system is largely composed of (a) AI+X Post and (b) Intelligent Surveillance Worker Nodes.

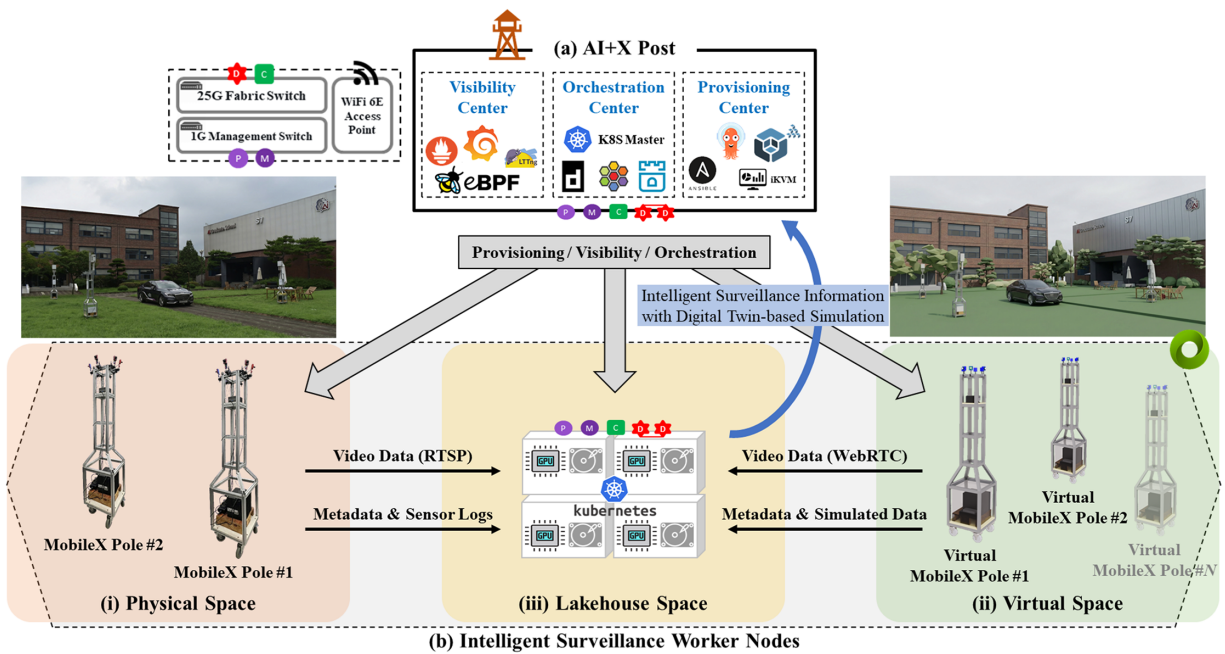


Figure 1: Overall architecture of the proposed scalable digital twin framework.

The AI+X Post in Fig. 1a is the top-level management and orchestration layer, which controls the resource management and execution flow of the entire system. The Intelligent Surveillance Worker Nodes in Fig. 1b are the execution layer that performs actual data collection, simulation, and analysis. These Worker Nodes are divided into three logical spaces: Physical Space in Fig. 1i, Virtual Space in Fig. 1ii, and Lakehouse Space in Fig. 1iii. Physical Space consists of the real-world environment where the actual surveillance target exists and the physical surveillance equipment that observes it. In this paper, we acquire video data for intelligent surveillance using the MobileX Pole developed by the authors. Virtual Space is a simulation space that recreates the physical surveillance environment in the form of a digital twin. This space allows for scalable surveillance scenarios and scale without limitations on the number of physical devices or installation environments. Lakehouse Space is a central processing space that integrates and collects, stores, and analyzes data generated from the Physical Space and Virtual Space. This space generates baseline data for quantitative performance evaluation. Each space has a different connection structure depending on the data characteristics and operating environment. Infrastructure comprised of fixed nodes, such as the AI+X Post and Lakehouse Space, is interconnected via a wired switch-based network for high bandwidth and low latency communication. However, mobile objects, such as MobileX Pole in Physical Space, connect to the network using short-range wireless communication based on Wi-Fi 6E and LTE/5G mobile networks. This structure forms a dual space digital twin, interconnecting the actual physical environment and the virtual simulation environment, which is designed to enable quantitative analysis of the system’s processing limits even as the number and scale of monitoring targets increase.

3.1 AI+X Post

The AI+X Post performs the following operations for applications appropriate for the individual Worker nodes: installation, configuration, deployment, control, and monitoring. It also serves as a central orchestration and control tower that interconnects and manages each distributed Intelligent Surveillance Worker Node. The core purpose of the AI+X Post is to provide control capabilities to ensure that physical and virtual workloads running on individual nodes are stable, repeatable, and scalable as needed. The AI+X Post hardware and software stack, as shown in Fig. 2, consists of (a) the Infrastructure Layer, (b) the Provisioning & Management Layer, (c) the Cloud-native Orchestration Layer, and (d) the Visibility & Analytics Layer.

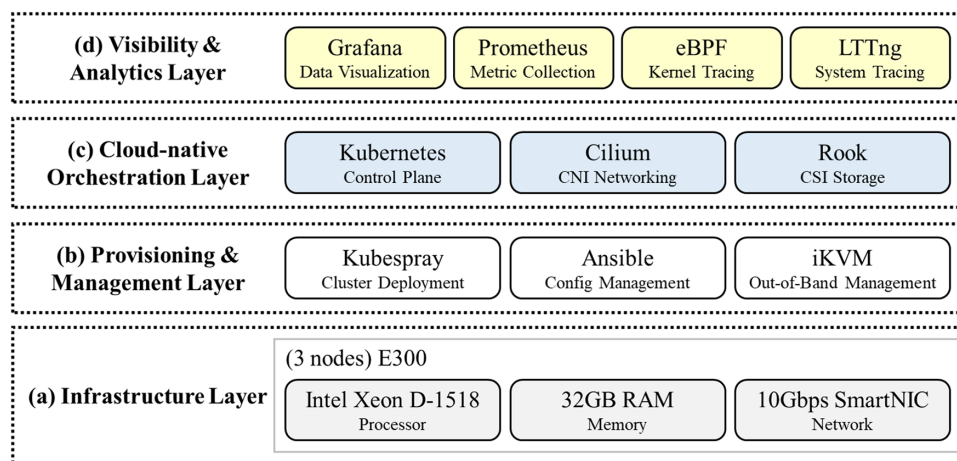


Figure 2: Hardware and software stack of the AI+X Post.

The Infrastructure Layer in Fig. 2a represents the physical hardware layer that constitutes the AI+X Post. The AI+X Post is built as an on-premises cluster consisting of three nodes, ensuring high availability.

Communication between these cluster nodes is connected via a 10 Gbps SmartNIC, which minimizes bottlenecks caused by management traffic and control messages. This layer provides the following foundational infrastructure for the stable operation of the upper-level software stack: compute, memory, and network resources. The Provisioning & Management Layer in Fig. 2b handles the initial configuration and ongoing operational management of the entire cluster. This layer performs the following functions: Kubespray is used to automatically deploy a Kubernetes cluster containing worker nodes, and Ansible consistently handles node configuration, package management, and configuration changes. It also enables remote recovery in the event of operating system failures or network outages through out-of-band management capabilities leveraging iKVM. This layer significantly alleviates the operational burden when conducting large-scale experiments and stress tests by ensuring rapid deployment of experimental environments and repeatable reproducibility.

The Cloud-native Orchestration Layer in Fig. 2c is the core control layer responsible for the deployment and execution of container-based workloads. The layer provides the following functions: It comprehensively manages container deployment, scheduling, and lifecycle management centered on Kubernetes, and provides high-performance eBPF-based CNI networking through Cilium. It leverages Rook to integrate distributed storage into the Kubernetes environment, which supports the stable execution of stateful workloads. This layer allows AI analytics containers and simulation instances to be dynamically scaled up or down based on experimental conditions, which allows for flexible resource utilization in an on-premises environment while maintaining cloud native characteristics. The Visibility & Analytics Layer in Fig. 2d is the visibility layer for observing and analyzing the overall system status. This layer collects node and container-level status information through Prometheus and visualizes it using Grafana. It tracks network and system events at a low level using eBPF-based kernel tracing and LTTng, which enables precise analysis of performance bottlenecks and abnormal behavior occurring in distributed environments. AI+X Post is built as an on-premises cluster environment that ensures high availability, which adopts a cloud-native architecture to support both scalability and operational efficiency. The physical resources provided by the Infrastructure Layer are configured in an automated and standardized manner, which the Cloud-native Orchestration Layer dynamically manages AI analytics and simulation workloads. The Visibility & Analytics Layer enables observation of the entire process across the three previous layers, which enables stable reproduction of identical conditions and precise analysis of performance changes in repetitive experimental environments.

3.2 Intelligent Surveillance Worker Nodes

Intelligent Surveillance Worker Nodes are the actual workload processing layer where actual surveillance data is collected, digital twin simulations are executed, and intelligent analytics are performed. This layer performs actual computations based on control commands transmitted from the AI+X Post. It is the area where large-scale video streams and simulation data are generated and intensively processed. This study adopted a structure that logically separates worker nodes based on data generation location and processing purpose, rather than simply configuring physical servers. Accordingly, Intelligent Surveillance Worker Nodes are composed of three spaces: (a) Physical Space, (b) Virtual Space, and (c) Lakehouse Space, as shown in Fig. 3. Each space has four layers: (i) Infrastructure Layer, (ii) System Software & Runtime Layer, (iii) Middleware & Engine Layer, and (iv) Application & Service Layer, which interact with each other.

The Space based architecture, illustrated in Fig. 3, is designed to separate data collection in real environments, large-scale simulations in virtual environments, and centralized analysis and storage into distinct execution areas. Each Space is comprised of four layers: (i) the Infrastructure Layer defines the physical resources where computation and storage are performed. (ii) The System Software & Runtime Layer provides the basic operating environment for executing workloads on those resources. (iii) The Middleware & Engine Layer handles core computations such as image processing, simulation execution, and AI inference.

(iv) The Application & Service Layer utilizes modules in the lower layers to perform application functions for surveillance and analysis. This hierarchical structure allows for the consistent management of Physical, Virtual, and Lakehouse Spaces, each with its own distinct characteristics, which allows for the load generated during stress testing to be clearly observed across the data generation, simulation, and analysis processing stages. By providing the flexibility to independently scale and adjust each space, the hierarchical structure creates an execution environment suitable for large-scale digital twin-based surveillance experiments. The roles of each space and the components of each layer are described in detail in the following Sections.

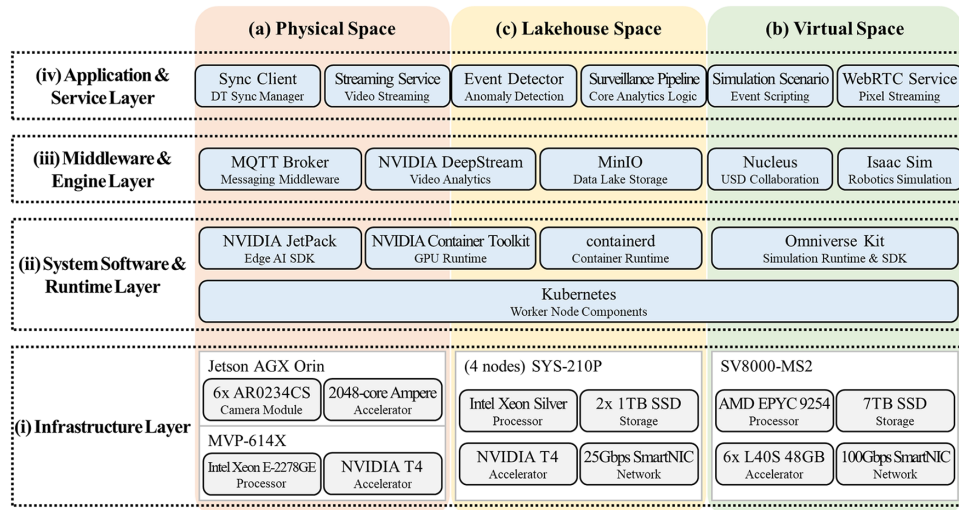


Figure 3: Hardware and software stack of the intelligent surveillance worker nodes.

3.2.1 Physical Space

The Physical Space is the execution space, encompassing the real-world environment where the actual surveillance target exists and the mobile edge surveillance nodes that observe it. This space is a real-world data generation area that generates image and sensor data acquired in real environments, providing reference input for digital twin-based stress testing. In the Physical Space shown in Fig. 3a, (i) the Infrastructure Layer consists of edge computing hardware that comprises the MobileX Pole. The proposed framework leverages a Jetson AGX Orin based platform to simultaneously capture six video streams from global shutter cameras of a single pole. When additional resources are required for data processing and AI workloads, the MVP-614X node equipped with NVIDIA T4 GPUs can be utilized for expanded resources. (ii) The System Software & Runtime Layer uses NVIDIA JetPack as the basic software stack for edge computing, and uses the NVIDIA Container Toolkit and containerd to run microservice-based GPU-accelerated workloads. Additionally, it receives commands from the Kubernetes master (i.e., control plane) across all three Spaces, controls the container runtime, and manages the pod status. (iii) The Middleware & Engine Layer performs preprocessing and management of six video streams using NVIDIA DeepStream, and MQTT Broker-based technology is applied for metadata exchange and digital twin synchronization. This layer acts as an intermediate layer between video frame processing and event and metadata generation, ensuring system scalability by processing video data and control messages separately. (iv) The Application & Service Layer runs a video streaming service and a Sync Client for digital twin synchronization. This layer transmits video and metadata generated in the Physical Space to the central Lakehouse Space via RTSP-based streaming, maintaining temporal and structural consistency with the digital twin environment. This four-layer architecture creates a consistent processing flow where multiple video streams generated from lower-level edge computing

resources are sequentially preprocessed and metadata encoded and then delivered to the central analytics space via the upper service layer. Physical Space reliably generates input data that reflects the characteristics of the real environment, while also serving as a reference workload for digital twin-based experiments.

3.2.2 Virtual Space

Virtual Space is a digital twin-based simulation space that extends beyond the physical space. It is a virtual execution layer that expands the scale of surveillance objects and experimental conditions without physical constraints. This space enables digital twin-based stress testing for large-scale surveillance scenarios by generating virtual sensing and rendering results that mimic the data structures collected in real environments. The infrastructure layer of the Virtual Space in Fig. 3i utilizes six L40S GPUs as hardware acceleration resources for the generation and rendering of high-load virtual workloads to build a realistic digital twin environment. It provides the computational power to simultaneously run multiple virtual surveillance nodes and high-resolution sensor simulations. System Software & Runtime Layer in Fig. 3ii comprises the foundational software stack for configuring a digital twin environment, centered around the NVIDIA Omniverse Kit. This layer provides a common execution environment for virtual object creation, sensor modeling, and rendering pipeline execution. It allows for a flexible definition of digital twin scenes that reflect the structure and behavioral characteristics of real-world surveillance environments. Middleware & Engine Layer in Fig. 3iii leverages NVIDIA Isaac Sim, which supports high-fidelity simulation based on physics laws, to simulate the behavior, sensor responses, and environmental interactions of surveillance objects within the virtual environment. Compared to other domain-specific simulators such as CARLA or AirSim, Isaac Sim was selected for its robust integration with containerized microservices and its native support for standardized 3D components (e.g., SimReady assets). This ensures that the virtual environment can be readily extended to incorporate more complex environmental elements in future scenario expansions without structural changes. Additionally, NVIDIA Nucleus acts as a collaboration hub that shares digital twin assets and simulation states, supporting scenario sharing and state synchronization across multiple virtual MobileX Pole instances. It enables consistent digital twin configuration even in distributed virtual testing environments. Application & Service Layer in Fig. 3iv executes the simulation scenario control module and WebRTC-based streaming service. This layer outputs information (i.e., video, events, and metadata) generated in the virtual environment in the same data structure as the Physical Space and transmits them to the central Lakehouse Space. This supports comparative experiments and integrated analysis between physical and virtual environments. This layered architecture forms a processing flow that manages and abstracts simulation results generated in a large-scale virtual surveillance environment and delivers them to a central analysis space. Thus, the virtual space enables scalable experiments that exceed the constraints of the physical environment, while also serving as a core load generation layer for digital twin-based stress testing.

3.2.3 Lakehouse Space

The Lakehouse Space aggregates video and metadata generated from the Physical Space and Virtual Space. It provides computing resources for AI analysis calculations for Intelligent Surveillance and stress test performance evaluations. This space provides resources capable of handling the large-scale input workloads generated by digital twin-based stress testing. It also supports and manages computing resources, converting these into quantitative metrics to assess the scalability and limitations of the system. The infrastructure layer of Lakehouse Space in Fig. 3i provides 6 TB of distributed storage across four servers to ensure high availability. It also supports the exchange of high-bandwidth video and AI training data through a SmartNIC based dual 25 Gbps fabric connection. This configuration provides a computational and storage environment to stably accommodate multiple video streams and analysis requests simultaneously flowing from both

the physical and virtual spaces. System Software & Runtime Layer in Fig. 3ii consists of the NVIDIA Container Toolkit and containerd for GPU-accelerated workload execution, and analysis tasks are managed as containers within a Kubernetes worker node environment. This allows the intelligent surveillance analytics pipeline to be deployed and scaled as an independent execution unit. It allows for flexible adjustment of analysis resources as experiment scale changes or load increases. Middleware & Engine Layer in Fig. 3iii integrates NVIDIA DeepStream and MinIO. DeepStream performs intelligent surveillance AI operations, such as object detection and event inference, on multiple video streams transmitted from physical and virtual spaces. Frame-level metadata and event information generated during this analysis process are stored in MinIO-based object storage. This structure simultaneously enables real-time analysis results utilization and long-term accumulation of experimental data, forming a data management foundation for repetitive experiments and post-analysis. Application & Service Layer in Fig. 3iv executes the Event Detector and Surveillance Pipeline. This layer synthesizes DeepStream based analysis results to produce key performance information from stress tests, such as event frequency, processing delay, and throughput. This information is then relayed to the AI+X Post for central control and comparative analysis of experimental results. This layered architecture aggregates data generated from distributed physical and virtual surveillance environments into a Lakehouse Space, forming a step-by-step processing flow that leads to intelligent surveillance AI analysis and performance information generation. Lakehouse Space functions as an execution space responsible for analysis and evaluation in digital twin-based stress testing.

3.3 Operational Workflow and Data Interaction

This Section describes the workflow in which the proposed hierarchical architecture interacts in an actual operational environment. The proposed framework is designed to clearly separate control flow and data flow, ensuring both management efficiency and data processing stability even in large-scale surveillance environments. Fig. 4 schematically illustrates the operational workflow and data interaction structure between the AI+X Post and the Intelligent Surveillance Worker Nodes.

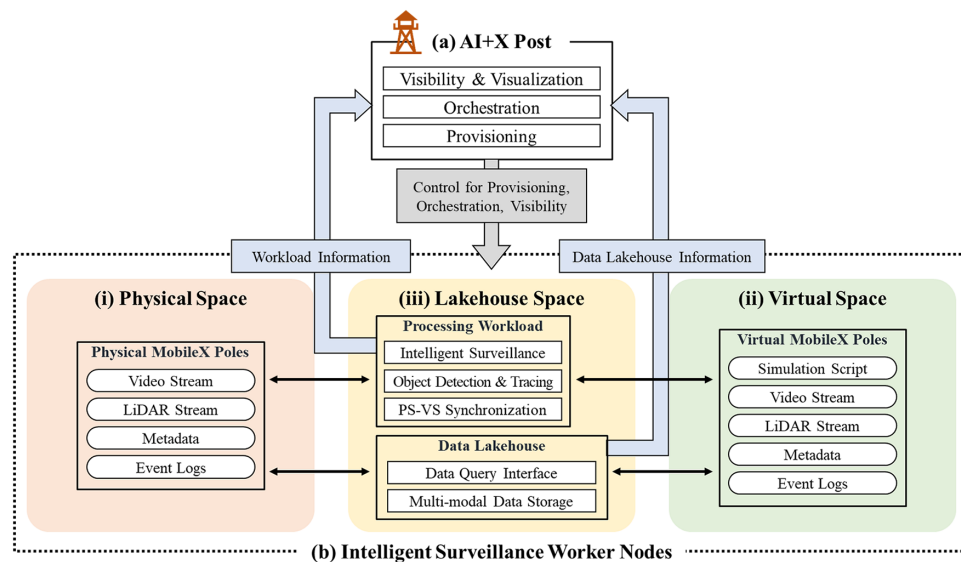


Figure 4: Operational workflow and data interaction of the proposed framework.

The overall system workflow is driven by a top-down control flow starting from the AI+X Post in Fig. 4a. The Provisioning and Orchestration modules within the AI+X Post direct the container deployment and

network configuration required for the Intelligent Surveillance Worker Nodes in Fig. 4b based on the requirements defined in the experimental scenario. This process's control signal transmission transitions each distributed node into a state ready for data collection and analysis. This means that the central AI+X Post consistently controls the configuration of the entire cluster, preventing configuration inconsistencies between individual nodes and ensuring a reproducible experimental environment.

When the system is activated, data interaction occurs between each space within the Intelligent Surveillance Worker Nodes. Real MobileX Poles in Physical Space of Fig. 4i and Virtual MobileX Poles in Virtual Space in Fig. 4ii generate video streams and metadata collected from the real and virtual environments, respectively. These heterogeneous data streams are transmitted in real-time to the central Lakehouse Space in Fig. 4iii. The Physical MobileX Pole utilizes RTSP and MQTT protocols, while the Virtual MobileX Pole utilizes WebRTC and Nucleus protocols to minimize transmission delays and transmit data to the Lakehouse Space.

The Processing Workload within the Lakehouse Space is the core computational engine that actually processes incoming data. The Processing Workload performs Intelligent Surveillance and Object Detection algorithms on streams received from the Physical MobileX Pole and Virtual MobileX Pole. In particular, the PS-VS Synchronization module within the Processing Workload is responsible for real-time state mirroring from the physical environment to the virtual digital twin. Rather than merely aligning offline data, this module processes continuous physical video streams via the RTSP Gateway and forwards them to the Virtual Space, where they are mapped as textures onto the virtual objects. This pipeline enables the system to measure end-to-end processing delays, ensuring that the virtual environment reflects the physical state within bounded latency. This synchronization serves as a prerequisite for conducting quantitative stress tests across the dual-space infrastructure.

The processed results data is utilized for two purposes. First, refined metadata and event logs are permanently stored in the multi-modal data storage within the Data Lakehouse, where they are utilized for post-analysis and model retraining via the Data Query Interface. Second, workload information, including real-time resource usage and analysis status, is relayed back to the higher level AI+X Post. Based on this information, AI+X Post's Visibility & Visualization module monitors the status of the entire cluster and forms a feedback loop that dynamically adjusts orchestration policies when load imbalances or anomalies occur. The continuous data processing and state mirroring flow executed within the Lakehouse Space is summarized in Algorithm 1.

Algorithm 1: Real-time physical-virtual state synchronization pipeline

Input: Physical RTSP video streams (S_p)

Output: Synchronized virtual environment and performance metrics

- 1: Initialize RTSP Gateway in Lakehouse Space
 - 2: while Intelligent Surveillance System is Active do
 - 3: //Step 1: Data Ingestion & Replication (Lakehouse Space)
 - 4: Receive physical frame f_p from S_p
 - 5: Record ingestion timestamp t_{in}
 - 6: Replicate f_p to simulate large-scale stream workload
 - 7: //Step 2: State Synchronization (Virtual Space)
 - 8: Transmit streams from Lakehouse Space to Virtual Space
 - 9: Decode f_p and execute texture mapping onto Virtual MobileX Poles
 - 10: Record rendering completion timestamp t_{out}
-

(Continued)

Algorithm 1 (continued)

```

11: //Step 3: Overhead Measurement & Logging
12: Calculate End-to-End Latency =  $t_{out} - t_{in}$ 
13: Measure current GPU VRAM consumption
14: Store latency and VRAM metrics to Data Lakehouse
15: end while

```

This workflow organically combines the centralized control of AI+X Post with the distributed data processing of Intelligent Surveillance Worker Nodes, ensuring operational stability and data integrity for a large-scale digital twin surveillance system that integrates physical and virtual spaces.

4 Experiments and Evaluation

This Section presents an integrated testbed environment built to verify the effectiveness of the proposed Scalable Digital Twin Framework, along with step-by-step stress test scenarios utilizing it. The purpose of this experiment was to verify the system's ability to reliably handle large-scale workloads in a complex surveillance environment combining physical and virtual environments, and to quantitatively analyze resource usage and performance bottlenecks as components scale. To achieve this, we designed three experimental scenarios, encompassing measurements of actual data generation, virtual environment scalability, and dual-space synchronization stability.

4.1 Experimental Setup

To evaluate the performance of the proposed framework, we built an actual hardware infrastructure based on the architecture designed in Section 3. The experimental environment is divided into the AI+X Post, the control plane, and the Intelligent Surveillance Worker Nodes (Physical, Lakehouse, and Virtual Space) on the data plane, based on their roles. The AI+X Post, comprised of three Supermicro E300-8D servers on-premises, is responsible for provisioning and orchestrating all experimental nodes and collecting real-time visibility to track resource status throughout the experiment. Physical Space generates real-world surveillance data using the NVIDIA Jetson AGX Orin, which performs edge computing, and MobileX Pole, which features the ADLINK MVP-614X. Lakehouse Space consists of four Supermicro SYS-210P servers, powered by a dual-25 Gbps network and NVIDIA T4 GPUs to perform real-time AI inference on large-scale streams. Virtual Space utilizes high-performance SV8000-MS2 servers and six NVIDIA L40S GPUs to handle high-precision physics simulations and rendering workloads. Detailed specifications of the entire experimental hardware are shown in Table 2.

Table 2: Detailed hardware specifications of the experimental testbed.

Components	AI+X Post	Lakehouse Space	Physical Space	Virtual Space
Server Model	Supermicro E300-8D (3 nodes)	Supermicro SYS-210P (4 nodes)	NVIDIA Jetson AGX Orin ADLINK MVP-614X	Leaders Systems SV8000-MS2
Processor	Intel Xeon D-1518 (4-Core, 2.20 GHz)	Intel Xeon Silver 4310 (12-Core, 2.1 GHz)	ARM Cortex-A78AE (12-Core, v8.2) Intel Xeon R-2278GE (8-Core, 3.3 GHz)	AMD EPYC 9254 (24-Core, Dual Socket)

(Continued)

Table 2 (continued)

Components	AI+X Post	Lakehouse Space	Physical Space	Virtual Space
Memory (RAM)	32 GB DDR4	32 GB DDR4	32 GB LPDDR5 32 GB DDR4	768 GB ECC DDR4
Accelerator	N/A	NVIDIA T4	Ampere GPU (2048 Cores) NVIDIA T4	6x NVIDIA L40S
Network Interface	Dual 10 GbE (Intel X552)	Dual 25 GbE (ConnectX-4 Lx)	Wi-Fi 6E Module (Intel AX200NGW)	100 GbE (ConnectX-5)
Storage	Total 1.5 TB NVMe SSD	Total 8 TB NVMe SSD	64 GB eMMC 5.1 1 TB NVMe SSD	7 TB NVMe SSD

4.2 Experimental Scenarios

To verify the performance and limitations of the proposed framework step by step, we designed experimental scenarios from three perspectives: data ingestion, virtual environment scalability, and real-time integration. All experiments were conducted using high-resolution video streams at FHD (1920×1080) with a frame rate of 65 FPS. This frame rate was selected as it represents the native maximum of the AR0234CS global shutter camera module used in the physical nodes. The configuration was designed to impose the highest possible data ingestion and rendering load, simulating high-density surveillance scenarios. The goal was to quantitatively identify system resource saturation points and performance bottlenecks at each step.

4.2.1 Scenario 1: Baseline Workload for Data Ingestion

The first scenario, as shown in Fig. 5a, identifies the workload characteristics occurring in the Physical Space and verifies the stable data ingestion and storage performance into the Lakehouse Space. To discuss the scalability of a virtual environment, a clear baseline must first be established regarding the load a single physical object imposes on the system. To achieve this, all six global shutter cameras mounted on a single Physical MobileX Pole were activated, performing high-quality video streaming at 1920×1080 resolution, 65 FPS, and H.264 codec (Target Bitrate: 9 Mbps CBR). The generated data is transmitted to the RTSP Gateway Server located within the Lakehouse Space via a Wi-Fi 6E wireless network and immediately pipelined to Data Lake storage. Throughput and packet loss rates were measured primarily between the Lakehouse Server's network interface (NIC) and the wireless AP, where data is ingested. These evaluations allow for the preliminary identification of signal instability in the wireless section and potential storage I/O bottlenecks during large-scale data collection. In addition, they allow for the development of a model of the network bandwidth required for large-scale expansion.

4.2.2 Scenario 2: Scalability Analysis of Virtual Space Simulation

The second scenario in Fig. 5b is a scalability test that analyzes the maximum number of objects the simulation engine can handle in a virtual space without physical constraints. In this step, we verify the system's capacity limitations for the data load generated by Virtual Space itself, without external data inflow. To achieve this, we adopted a method of gradually increasing the number of Virtual MobileX Pole instances (N) based on NVIDIA Isaac Sim within Virtual Space from 0 to the maximum. Each virtual pole generates six independent 1080p/65fps video streams and performs physics engine and rendering operations, identical

to the baseline load derived from Scenario 1. Key metrics include real-time tracking of VRAM usage (GiB) and GPU utilization (Utilization) as instances increase on the NVIDIA L40S GPU, which handles simulation and rendering operations. Based on the collected data, the initial fixed cost (i.e., cold-start cost) for building the simulation and the marginal cost for adding objects are modeled using formulas. Finally, the maximum monitoring capacity that a single GPU and the entire cluster can support without performance degradation is calculated to determine the system's scalability limits.

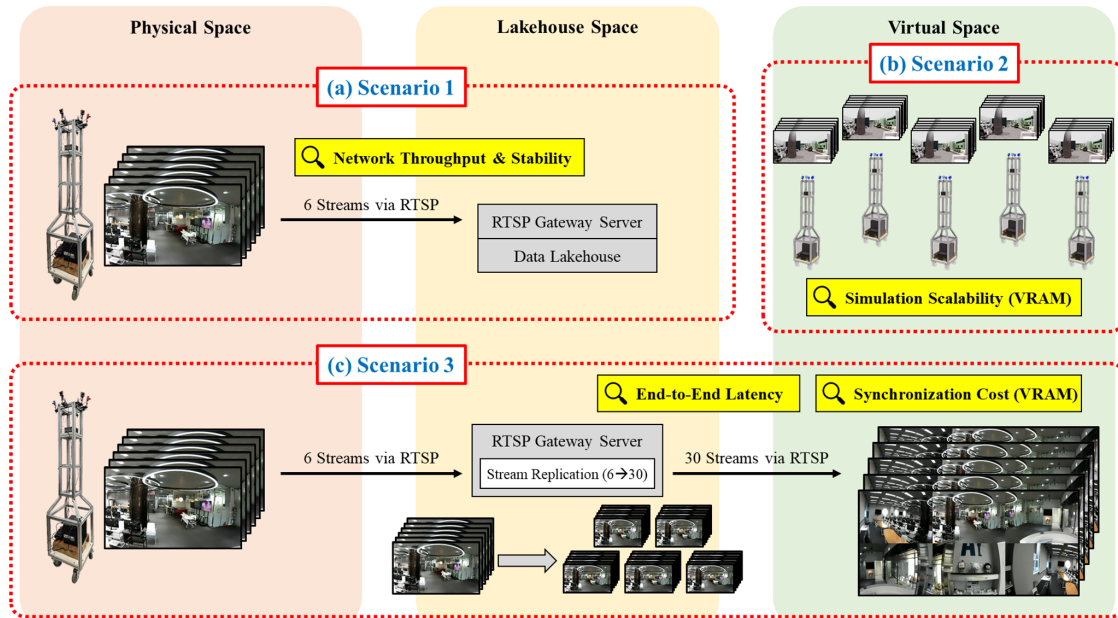


Figure 5: Overview of the three experimental scenarios evaluating network throughput (Scenario 1), simulation VRAM scalability (Scenario 2), and end-to-end synchronization latency (Scenario 3).

4.2.3 Scenario 3: Overhead Analysis of Dual-Space Synchronization

The final scenario, as depicted in Fig. 5c, verifies the cost and reliability incurred when real-time video streams from the Physical Space are synchronized to the Virtual Space via the Lakehouse Space. Digital twins must go beyond simply generating virtual data and mirror real-world data without delay. To verify this, a data pipeline scenario is required, where video streams transmitted from the Physical Space are relayed through the Lakehouse Space's RTSP Gateway and then mapped as textures to screens within the Virtual Space's digital twin. To simulate load conditions exceeding the physical equipment capacity, the input stream was virtually replicated in the Lakehouse Space, increasing the load by simultaneously decoding and rendering it in the Virtual Space. The measurement metrics include the system processing latency from the time the video reaches the RTSP Gateway until it is rendered on the Virtual Space screen, as well as the additional VRAM overhead consumed when decoding external video. It validates the system's guaranteed response speed (i.e., latency) and operational efficiency when large-scale control data is integrated in real-time, demonstrating the effectiveness of the proposed framework in a real-world control environment.

4.3 Experimental Results and Analysis

This section quantitatively analyzes the system's resource efficiency and scalability limitations based on experimental results for the three defined scenarios. All data used average values collected after the system reached operational stability.

4.3.1 Result of Scenario 1

In the first experiment, the network traffic generated by a single Physical MobileX Pole (i.e., six video streams) was measured 500 times. Each trial was performed with 10 min of continuous video streaming, resulting in an average total throughput of 58.09 Mbps (SD = 0.83), as shown in Fig. 6. This stream size represents only about 0.12% of the dual 25 GbE bandwidth of the wired network supported by Lakehouse Space, confirming no data collection bottlenecks in the wired section. In contrast, the analysis of the Wi-Fi 6E wireless segment reveals that it constitutes the main bottleneck for system expansion. In a line-of-sight (LoS) environment, sufficient network bandwidth was available, stably meeting the required throughput of 58.09 Mbps. However, in a non-line-of-sight (NLoS) environment, where obstacles such as walls are present, the average throughput dropped to 48.36 Mbps (SD = 3.15), resulting in data loss. In particular, in the NLoS environment, throughput fluctuated significantly, dropping to a low of 38.50 Mbps momentarily. Although single-pole traffic (58.09 Mbps) in a LoS environment can be handled reliably, the rapid throughput fluctuations observed in an NLoS environment can potentially cause packet jitter when multiple poles transmit data simultaneously. Therefore, for future large-scale expansion, it was confirmed that not only a simple bandwidth increase, but also adaptive jitter buffering optimization at the edge and precise wireless coverage design that minimizes shadow areas are prerequisites for ensuring system stability.

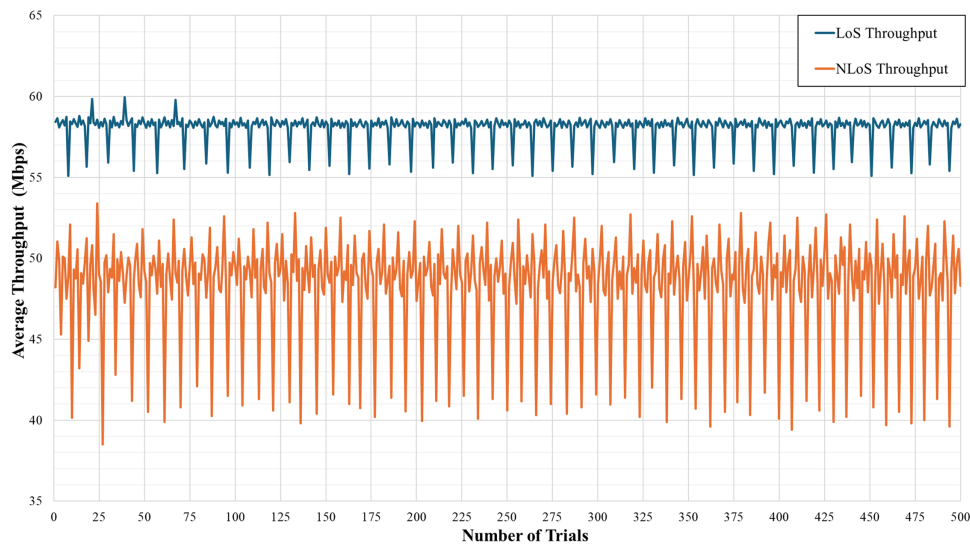


Figure 6: Aggregate network throughput per trial for 6 FHD streams over 500 trials under LoS and NLoS conditions.

4.3.2 Result of Scenario 2

This experiment focused on predicting the scalability of the entire system by analyzing resource consumption patterns (i.e., resource consumption during virtual pole rendering) in a single-GPU environment. Fig. 7 shows the change in VRAM usage (GiB) as the number of virtual MobileX Pole instances (N) increases. Each dot represents the average VRAM usage calculated from 10 repeated evaluations under the same conditions. The dotted line intersecting the dots in Fig. 7 represents a linear regression model that reflects the data's trends.

The evaluation results reveal three key characteristics: First, a high initial fixed cost was observed. Even without deploying any virtual poles ($N = 0$), Virtual Space was found to consume approximately 12.85 GiB (SD = 0.12) of VRAM. It can be explained by the essential system overhead required to initialize the high-precision physics engine (PhysX), configure the rendering pipeline, and load background assets. This initial

cost is an opportunity cost for ensuring the fidelity of the simulation, but as the number of instances increases, its proportion of total resources gradually decreases. Second, there is clear linear scalability. With each additional instance, VRAM usage consistently increased by approximately 1.06 GiB per instance, modeled by a linear regression equation of $VRAM(N) \approx 1.06N + 12.85$. The analysis results showed a coefficient of determination (R^2) of over 0.999, with a 95% confidence interval for the slope of [1.063, 1.067], indicating predictable resource allocation. This high degree of linearity arises because, when assets with identical structures are allocated in GPU memory, their resource consumption follows an architecturally consistent pattern. This means that the required resource amount can be reliably estimated without sudden load surges. Third, the maximum capacity is determined by hardware limitations. The NVIDIA L40S GPU used in the experiment has 48 GiB of physical memory, but due to the ECC (Error Correction Code) feature enabled for datacenter-grade reliability, the actual available memory is limited to approximately 45 GiB (46,068 MiB). According to the analyzed linear model, when $N = 30$, VRAM usage reaches approximately 44.60 GiB, approaching the available threshold of 45 GiB. Therefore, the safe maximum capacity in a single-GPU environment was confirmed to be 30 Virtual MobileX Poles (i.e., a total of 180 video streams). In conclusion, we experimentally demonstrated that the proposed framework’s full cluster consisting of six L40S GPUs can simulate up to 180 MobileX Poles and 1080 FHD real-time video streams without performance degradation through workload distribution.

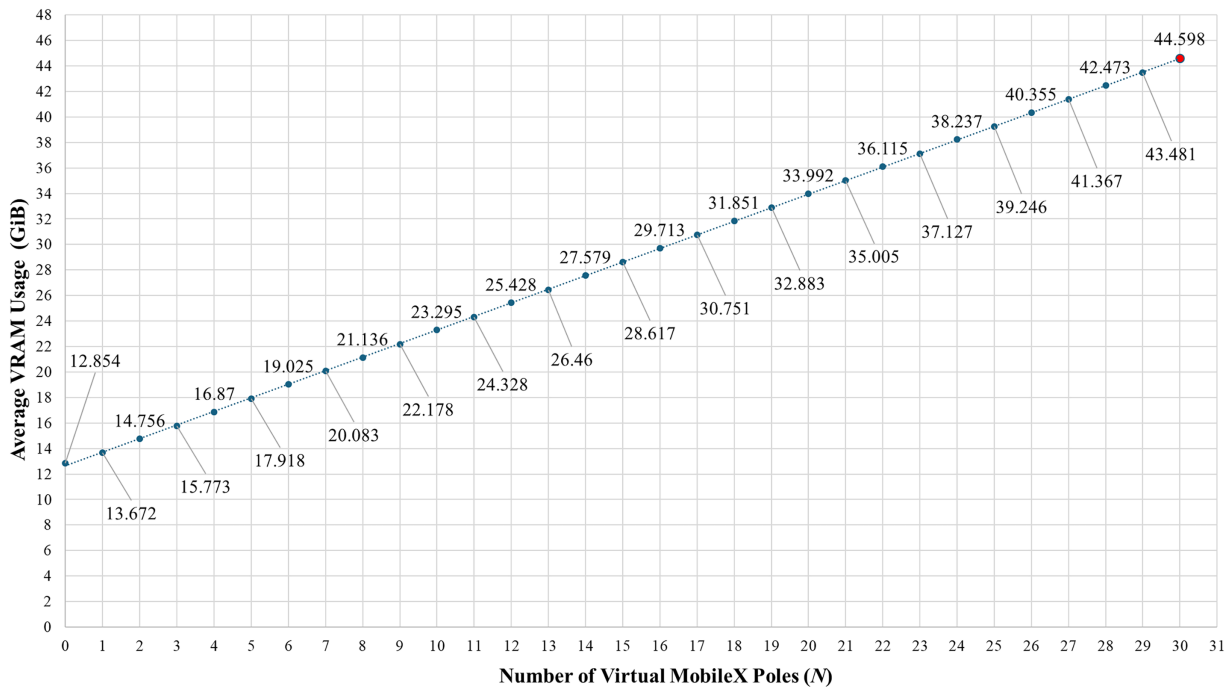


Figure 7: Average VRAM usage for virtual MobileX Poles ($N = 0-30$) across 10 trials, with linear scalability model ($VRAM(N) \approx 1.06N + 12.85$).

4.3.3 Result of Scenario 3

The final experiment verified the framework’s performance during the synchronization of video data from the Physical Space with the digital twin. Fig. 8 shows the changes in system latency and VRAM overhead when processing streams replicated from one Physical MobileX Pole to five Virtual MobileX Pole streams (i.e., 30 streams in total) over a total of 500 evaluations. The delay time defined in this experiment refers to the elapsed time from the time when the RTSP Gateway Server in the Lakehouse Space replicates and

transmits the received RTSP stream to the time when the Virtual Space receives and decodes the frame and completes texture mapping.

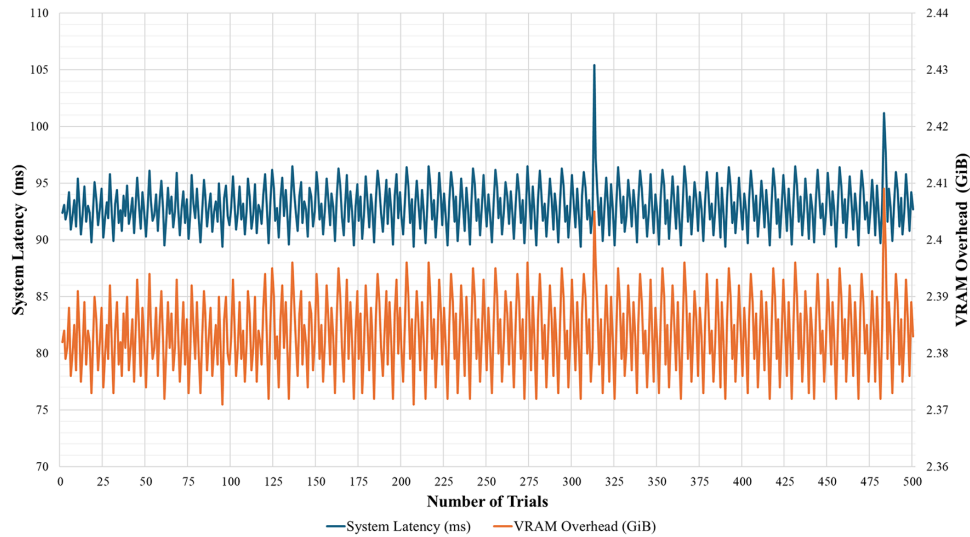


Figure 8: System latency and VRAM overhead per trial for 30 replicated FHD streams over 500 trials.

The evaluation results showed that the average latency required to synchronize 30 FHD (65 FPS) streams in a 100 GbE-based network environment was 92.95 ms (SD = 1.96). The blue line in Fig. 8, which shows system latency, maintained a stable value between 90 and 95 ms for most of the time, but intermittent spikes were observed in certain periods (e.g., Trials 313 and 483). These intermittent spikes are analyzed as being caused by temporary buffering during the processing of high-bandwidth data (e.g., I-frames) and the execution of OS background tasks on the receiving-side (Virtual Space) system. Even during these spikes, peak latency was suppressed to approximately 105 ms. This comfortably satisfies the stringent latency constraints typically required for real-time AI-assisted monitoring and remote teleoperation, which generally demand end-to-end delays below 100 to 200 ms to prevent operational performance degradation [46]. In terms of resource efficiency, the average VRAM usage required to display 30 replicated video streams in the virtual space was measured at approximately 2.38 GiB (i.e., orange line). This results in a memory overhead of approximately 487 MiB per MobileX Pole (i.e., 6 streams). It is noteworthy that during periods of rapid latency spikes, VRAM usage also shows a positive correlation, increasing in sync. When a bottleneck occurs on the receiving side, the proposed framework accurately reflects the physical characteristics of the system, where frames coming from Lakehouse Space are temporarily accumulated in the ring buffer on GPU VRAM. In conclusion, the proposed framework leverages the stream replication and distribution capabilities of the Lakehouse Space to demonstrate that stable, latency-free synchronization can be achieved with only approximately a 45% additional overhead compared to the pure simulation workload (approximately 1.06 GiB per unit), even when integrating large-scale data from the physical environment into the virtual space.

5 Conclusion

This paper proposed a stress testing method to quantitatively analyze the scalability limitations and computing resource consumption characteristics of a digital twin for intelligent mobile surveillance systems. The study designed and implemented a scalable digital twin framework based on a real MobileX Pole, capable of integrating physical sensor streams and large-scale virtual simulations. The proposed framework and quantitative stress testing method were validated by evaluating network throughput and GPU VRAM

usage as the number of real and virtual mobile poles increased. Experimental results demonstrated that high-fidelity digital twin simulations incurred high fixed costs during initial rendering and physics engine initialization, whereas the additional resource consumption resulting from an increase in the number of agents exhibits an almost linear behavior. In the experimental environment, the framework was shown to stably support up to 180 virtual MobileX Pole instances, corresponding to a total of 1080 FHD virtual video streams. Moreover, the overhead associated with synchronizing real-time video streams from the physical space to the virtual environment remained at a manageable level, demonstrating the practical implementability of large-scale digital twin-based surveillance systems with a physical-virtual integrated architecture. Rather than focusing solely on the visual fidelity of digital twins or the performance of individual surveillance algorithms, this study quantitatively characterized the system-level scalability and resource constraints of digital twin-based mobile surveillance infrastructures. In particular, through an experimental analysis of the effects of virtual environment scaling and real sensor stream integration on system resources, this work presented practical design criteria for surveillance systems built upon high density visual data and digital twin environments.

While the proposed framework demonstrated strong scalability in large-scale virtual environments, the physical experiments were limited to data ingestion from a single MobileX Pole. Although this setup was sufficient to validate the fundamental physical-virtual synchronization pipeline, it does not fully reflect the complex network dynamics, interference, and distributed overheads expected in multi-node deployments. In addition, although linear VRAM scalability was observed on a single GPU node, extending the framework to distributed multi-GPU clusters is likely to introduce additional synchronization overhead. Future work will focus on extending the framework to more realistic scenarios, including $N:N$ synchronization across multiple distributed physical nodes and large-scale virtual environments spanning multiple GPU clusters. Furthermore, the proposed stress testing method will be expanded to incorporate complex and time-critical conditions, such as dynamic events, failure propagation, and intensive distributed data processing.

Acknowledgement: The authors gratefully acknowledge the support of the AI Graduate School at the Gwangju Institute of Science and Technology (GIST) and the Korean government for enabling this research.

Funding Statement: This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant funded by the Ministry of Land Infrastructure and Transport (Grant RS-2023-00256888). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01842, Artificial Intelligence Graduate School Program (GIST)). This work was supported by the Technology Innovation Program (RS-2025-25448249, E2E Autonomous Driving Reference Data Construction and Core Technology Development) funded by the Ministry of Trade, Industry & Resources (MOTIR, Korea). This research was supported by the National Research Council of Science & Technology (NST) grant funded by the Korea government (MSIT) (No. GTL25041-000).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization & methodology, Sun Park and JongWon Kim; software, DongHwan Ku; validation, DongHwan Ku; investigation, DongHwan Ku; writing—review and editing, DongHwan Ku and Sun Park; visualization, DongHwan Ku; supervision, DongHwan Ku, Sun Park, and JongWon Kim; project administration, JongWon Kim; funding acquisition, JongWon Kim. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bafail O. Optimizing smart city strategies: a data-driven analysis using random forest and regression analysis. *Appl Sci.* 2024;14(23):11022. doi:10.3390/app142311022.
2. Tsybina E, Lebakula V, Zhang F, Hu Q, Laskey KB. Smart cities: the data to decisions process. *Nat Cities.* 2025;2(2):135–43. doi:10.1038/s44284-024-00194-7.
3. Myagmar-Ochir Y, Kim W. A survey of video surveillance systems in smart city. *Electronics.* 2023;12(17):3567. doi:10.3390/electronics12173567.
4. Khanpour A, Wang T, Vahidi-Shams A, Ectors W, Nakhaie F, Taheri A, et al. UAV-based intelligent traffic surveillance system: real-time vehicle detection, classification, tracking, and behavioral analysis. arXiv:2509.04624. 2025.
5. Ahmad T, Morel A, Cheng N, Palaniappan K, Calyam P, Sun K, et al. Future UAV/drone systems for intelligent active surveillance and monitoring. *ACM Comput Surv.* 2026;58(2):1–37. doi:10.1145/3760389.
6. Ku D, Park S, Kim J. Design and implementation of mobile SmartX pole for V2X communication data collection of cloud-native edge cluster. In: *Proceedings of Symposium of the Korean Institute of Communications and Information Sciences; 2024 Jun 19–22; Ramada Plaza Jeju, Republic of Korea.* (In Korean).
7. Ku D, Zang H, Yusupov A, Park S, Kim J. Vehicle-to-everything-car edge cloud management with development, security, and operations automation framework. *Electronics.* 2025;14(3):478. doi:10.3390/electronics14030478.
8. Park S, Kim J. Design of virtual driving test environment for collecting and validating bad weather SiLS data based on multi-source images using DCU with V2X-car edge cloud. *Comput Mater Contin.* 2026;86(3):15. doi:10.32604/cmc.2025.072865.
9. James P, Jonczyk J, Smith L, Harris N, Komar T, Bell D, et al. Realizing smart city infrastructure at scale, in the wild: a case study. *Front Sustain Cities.* 2022;4:767942. doi:10.3389/frsc.2022.767942.
10. Kumar V, Gunner S, Spyridopoulos T, Vafeas A, Pope J, Yadav P, et al. Challenges in the design and implementation of IoT testbeds in smart-cities: a systematic review. arXiv:2302.11009. 2023.
11. Forkan ARM, Kang YB, Marti F, Banerjee A, McCarthy C, Ghaderi H, et al. AIoT-CitySense: AI and IoT-driven city-scale sensing for roadside infrastructure maintenance. *Data Sci Eng.* 2024;9(1):26–40. doi:10.1007/s41019-023-00236-5.
12. Haag S, Anderl R. Digital twin—proof of concept. *Manuf Lett.* 2018;15(3):64–6. doi:10.1016/j.mfglet.2018.02.006.
13. El-Agamy RE, Sayed HA, AL Akhatatneh AM, Aljohani M, Elhosseini M. Comprehensive analysis of digital twins in smart cities: a 4200-paper bibliometric study. *Artif Intell Rev.* 2024;57(6):154. doi:10.1007/s10462-024-10781-8.
14. Liu W, Lv Y, Wang Q, Sun B, Han D. A systematic review of the digital twin technology in buildings, landscape and urban environment from 2018 to 2024. *Buildings.* 2024;14(11):3475. doi:10.3390/buildings14113475.
15. Sacoto-Cabrera EJ, Perez-Torres A, Tello-Oquendo L, Cerrada M. IoT, AI, and digital twins in smart cities: a systematic review for a thematic mapping and research agenda. *Smart Cities.* 2025;8(5):175. doi:10.3390/smartcities8050175.
16. Sudhakar S, Hanzelka J, Bobillot J, Randhavane T, Joshi N, Vineet V. Exploring the Sim2Real gap using digital twins. In: *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France.* doi:10.1109/ICCV51070.2023.01867.
17. Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun ACM.* 2022;65(1):99–106. doi:10.1145/3503250.
18. Park JS, O'Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: interactive simulacra of human behavior. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology; 2023 Oct 29–Nov 1; San Francisco, CA, USA.* doi:10.1145/3586183.3606763.
19. Xu H, Omिताomu F, Sabri S, Zlatanova S, Li X, Song Y. Leveraging generative AI for urban digital twins: a scoping review on the autonomous generation of urban data, scenarios, designs, and 3D city models for smart city advancement. arXiv:2405.19464. 2024.
20. Duran K, Shin H, Duong TQ, Canberk B. GenTwin: generative AI-powered digital twinning for adaptive management in IoT networks. *IEEE Trans Cogn Commun Netw.* 2025;11(2):1053–63. doi:10.1109/TCCN.2025.3527719.

21. Aminiyeganeh K, Coutinho RWL, Boukerche A. IoT video analytics for surveillance-based systems in smart cities. *Comput Commun.* 2024;224(4):95–105. doi:10.1016/j.comcom.2024.05.021.
22. Fei L, Han B. Multi-object multi-camera tracking based on deep learning for intelligent transportation: a review. *Sensors.* 2023;23(8):3852. doi:10.3390/s23083852.
23. Amosa TI, Sebastian P, Izhar LI, Ibrahim O, Ayinla LS, Bahashwan AA, et al. Multi-camera multi-object tracking: a review of current trends and future advances. *Neurocomputing.* 2023;552(7):126558. doi:10.1016/j.neucom.2023.126558.
24. Haq MA. Mobile surveillance system using unmanned aerial vehicle for aerial imagery. *Emerg Inf Sci Technol.* 2024;5(2):52–9. doi:10.18196/eist.v5i2.24837.
25. Choi Y, Kim H. Obstacle-aware crowd surveillance with mobile robots in transportation stations. *Sensors.* 2025;25(2):350. doi:10.3390/s25020350.
26. Saketh M, Nandal N, Tanwar R, Reddy BP. Intelligent surveillance support system. *Discov Internet Things.* 2023;3(1):9. doi:10.1007/s43926-023-00039-0.
27. Fuller A, Fan Z, Day C, Barlow C. Digital twin: enabling technologies, challenges and open research. *IEEE Access.* 2020;8:108952–71. doi:10.1109/ACCESS.2020.2998358.
28. Liu X, Jiang D, Tao B, Xiang F, Jiang G, Sun Y, et al. A systematic review of digital twin about physical entities, virtual models, twin data, and applications. *Adv Eng Inform.* 2023;55:101876. doi:10.1016/j.aei.2023.101876.
29. Huzzat A, Anpalagan A, Khwaja AS, Woungang I, Alnoman AA, Pillai AS. A comprehensive review of digital twin technologies in smart cities. *Digit Eng.* 2025;4(19):100040. doi:10.1016/j.dte.2025.100040.
30. Yessief M, Hakam Y, Tabaa M, Alammam MM, Elbarbary ZMS. Digital twin technology in smart cities: a step toward intelligent urban management. *Energy Rep.* 2025;14(Supplement 16):5539–57. doi:10.1016/j.egy.2025.11.097.
31. Li D, Yu W, Shao Z. Smart city based on digital twins. *Comput Urban Sci.* 2021;1(1):4. doi:10.1007/s43762-021-00005-y.
32. Wang H, Chen X, Jia F, Cheng X. Digital twin-supported smart city: status, challenges and future research directions. *Expert Syst Appl.* 2023;217:119531. doi:10.1016/j.eswa.2023.119531.
33. Lv Z, Chen D, Lv H. Smart city construction and management by digital twins and BIM big data in COVID-19 scenario. *ACM Trans Multimedia Comput Commun Appl.* 2022;18(2s):1–21. doi:10.1145/3529395.
34. White G, Zink A, Codecá L, Clarke S. A digital twin smart city for citizen feedback. *Cities.* 2021;110(1):103064. doi:10.1016/j.cities.2020.103064.
35. Lam HK, Lam PD, Ok SY, Lee SH. Digital twin smart city visualization with MoE-based personal thermal comfort analysis. *Sensors.* 2025;25(3):705. doi:10.3390/s25030705.
36. Kerbl B, Kopanas G, Leimkuehler T, Drettakis G. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans Graph.* 2023;42(4):1–14. doi:10.1145/3592433.
37. Fan B, Su Z, Chen Y, Wu Y, Xu C, Quek TQS. Ubiquitous control over heterogeneous vehicles: a digital twin empowered edge AI approach. *IEEE Wirel Commun.* 2023;30(1):166–73. doi:10.1109/mwc.012.2100587.
38. Barbuto V, Savaglio C, Lee EA, Fortino G. Engineering opportunistic digital twins with Lingua franca. *Future Gener Comput Syst.* 2026;178(10):108262. doi:10.1016/j.future.2025.108262.
39. Rahman MA, Shahrir MF, Iqbal K, Abushaiba AA. Enabling intelligent industrial automation: a review of machine learning applications with digital twin and edge AI integration. *Automation.* 2025;6(3):37. doi:10.3390/automation6030037.
40. Tang Z, Zhuang D, Zhang J. Evaluation framework for domain-specific digital twin platforms. *Sci Rep.* 2025;15(1):10544. doi:10.1038/s41598-024-82154-8.
41. Santos R, Piqueiro H, Dias R, Rocha CD. Transitioning trends into action: a simulation-based digital twin architecture for enhanced strategic and operational decision-making. *Comput Ind Eng.* 2024;198(1):110616. doi:10.1016/j.cie.2024.110616.
42. de M Del Esposte A, Santana EFZ, Kanashiro L, Costa FM, Braghetto KR, Lago N, et al. Design and evaluation of a scalable smart city software platform with large-scale simulations. *Future Gener Comput Syst.* 2019;93(6):427–41. doi:10.1016/j.future.2018.10.026.

43. Samak TV, Samak CV, Binz J, Smereka J, Brudnak M, Gorsich D, et al. Off-road autonomy validation using scalable digital twin simulations within high-performance computing clusters. *arXiv:2405.04743*. 2024.
44. Nasrazadani H, Nogal M, Adey BT, Mitoulis SA. Prioritizing simulation-based stress tests to assess the resilience of transport systems: a computation-free methodology. *J Infrastruct Preserv Resil*. 2025;6(1):16. doi:10.1186/s43065-025-00128-0.
45. Li H, Balasubramanian P, Meiers M, Li J, Kaufmann A. SplitSim: large-scale simulations for evaluating network systems research. *arXiv:2402.05312*. 2024.
46. Kamtam SB, Lu Q, Bouali F, Haas OCL, Birrell S. Network latency in teleoperation of connected and autonomous vehicles: a review of trends, challenges, and mitigation strategies. *Sensors*. 2024;24(12):3957. doi:10.3390/s24123957.