EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

Edge-Cloud Collaborative Digital Twin and Hierarchical Intelligent Decision-Making Model for Low-Latency Energy Regulation

Xiaohua Zou[1, *]

[1]ChangZhou University of Information Technology, Changzhou, China

Abstract

High penetrations of distributed energy resources require energy regulation that combines cloud-level global optimization with edge-level fast response. This paper proposes EC-HDT, a device-edge-cloud hierarchical digital twin in which a lightweight graph-attention-temporal-convolution estimator reconstructs local states under asynchronous, noisy, and missing measurements, while a cloud predictor and model predictive controller perform rolling economic optimization. A five-factor decision weight based on communication latency, information freshness, estimation confidence, operational risk, and edge computational load continuously allocates control authority between edge and cloud, and a quadratic-programming safety layer enforces physical constraints. On the IEEE 33-bus system, EC-HDT achieves a nodal-voltage MAE of 0.0076 p.u., mean/P95 end-to-end latencies of 56.4/89.4 ms, and a 99.2% control success rate; the daily operating cost is 3.51% lower than that of the fixed-fusion scheme. The results indicate that state-aware edge-cloud coordination can improve the latency-economy-safety trade-off in distribution-system regulation.

Keywords: digital twin; edge-cloud collaboration; energy regulation; state estimation; hierarchical decision-making; model predictive control

Received on 05 August 2026, accepted on 25 August 2026, published on 02 September 2026

doi: 10.4108/ew.14340

*Corresponding author. Email: xhua.zou@163.com

1. Introduction

The increasing penetration of photovoltaic generation, wind power, energy storage, and flexible loads is transforming distribution systems from traditional unidirectional supply networks into complex cyber-physical systems characterized by strong randomness, bidirectional power flow, and multi-timescale coupling. Conventional energy management typically uploads multi-source data to the cloud for centralized state estimation and optimal scheduling. Although global models and abundant computing resources can provide high-quality system-level decisions, communication queuing, network jitter, and cloud-side optimization increase closed-loop response time. Conversely, fully edge-autonomous control can reduce latency but cannot readily access cross-regional power exchanges, long-term storage constraints, or global electricity-price information. Therefore, low-latency energy regulation should not simply choose between edge and cloud computing; instead, computing and decision responsibilities should be dynamically allocated according to system-state changes, communication quality, and risk level. Digital twins provide a unified framework for online monitoring, state evolution, and closed-loop control by continuously mapping physical assets, real-time data, and virtual models. Existing reviews indicate that research on digital twins for smart energy systems is shifting from offline model construction toward real-time synchronization, model management, computing-resource allocation, and trustworthy control [1]. Studies that construct local-energy-system twins using real smart-meter data have also demonstrated the feasibility of data-driven replication and operational analysis [2] but closed-loop control and end-to-end latency remain insufficiently addressed.

Existing studies related to this problem can be broadly categorized into digital-twin architectures, edge-cloud collaborative energy management, and AI-based state estimation. Digital-twin research builds on foundational physical–virtual coupling and lifecycle-oriented digital models [3,4], and has subsequently expanded to intelligent energy applications [5]. Sifat et al. proposed a systems-engineering-based digital-twin grid framework emphasizing component modeling and lifecycle consistency [6], while Krishnan et al. demonstrated the feasibility of coordinating real-time edge control with cloud optimization in a real microgrid [7]. Digital twins have also been applied to microgrid cost optimization and hybrid MPC-based energy management [8,9]. However, these approaches generally rely on fixed deployment patterns and control periods, limiting their adaptability to network congestion and variations in state freshness.

Edge-cloud collaboration builds on edge computing and mobile edge computing (MEC), which move latency-sensitive computation closer to data sources while coordinating communication and computing resources [10,11]; Li et al. introduced federated dueling deep Q-network (DQN) into edge-cloud microgrid energy management [12]; and Habib et al. demonstrated the real-time value of local computing for complex battery control through an edge-based battery energy management system [13]. In AI-based state estimation, Kundacina et al. constructed a factor-graph GNN to approximate linear PMU state estimation [14]; Ngo et al. integrated Kalman-filter physical knowledge with a spatiotemporal GNN to improve estimation under limited measurements [15]; Cibaku et al. used graph attention to enhance robustness against false-data injection [16]; Yu and Wei proposed a spatial-temporal attention graph-based network (ST-AGNet) for dynamic power-system state prediction using adaptive graphs and multiscale attention [17]; and Raghuvamsi et al. combined physical constraints with a temporal convolutional network (TCN) for distribution-system state estimation under missing data[18]. Recent studies combining online learning with MPC have further shown that embedding predictive models into controllers can adapt to dynamic electricity prices, loads, and renewable-energy profiles [19]. MPC studies for alternating-current/direct-current (AC/DC) microgrids have also verified the suitability of rolling optimization for coupled storage and demand-response constraints [20].

Despite these advances, four research gaps remain. First, digital twins are often applied independently to monitoring, simulation, state estimation, or optimization rather than being organized as a unified closed-loop framework that spans multi-source sensing, edge-side state estimation, cloud-based prediction, decision coordination, and safe physical execution. Second, existing edge-cloud task allocation and control schemes are commonly based on fixed deployment patterns, fixed control periods, or predetermined fusion strategies, which prevents communication latency and state freshness from directly influencing control authority in real time. Third, although data-driven state-estimation methods have substantially improved reconstruction accuracy under noisy, asynchronous, and incomplete measurements, estimation confidence and operational risk are still weakly coupled with edge-cloud decision-making and physical safety constraints. Fourth, many existing studies focus primarily on operating cost, prediction accuracy, or state-estimation error, while mean latency, tail latency, communication overhead, and control success under abnormal communication and operating conditions are not jointly evaluated. These limitations indicate that an effective low-latency energy-regulation framework should not only improve individual estimation or optimization modules, but also coordinate state awareness, communication conditions, edge-cloud control authority, and safety constraints within a unified closed loop.

To address these gaps, this paper proposes an edge-cloud collaborative hierarchical digital twin model, termed EC-HDT. The main contributions are summarized as follows:

(1) A three-tier device-edge-cloud dual-timescale digital-twin architecture is developed, combining millisecond-level edge-side synchronization and fast control with cloud-based global prediction, model calibration, and rolling optimization.

(2) A lightweight graph-attention-temporal-convolution state-estimation network is developed for asynchronous, noisy, and partially missing measurements, jointly estimating system states, operational risk, and estimation confidence.

(3) A latency-aware hierarchical decision mechanism dynamically coordinates edge and cloud control according to communication latency, state freshness, estimation confidence, and operational risk, enabling adaptive transitions among cloud-dominant, edge-cloud fusion, and edge-autonomous modes.

(4) A closed-loop control framework combining cloud-based MPC and safety projection is established to jointly optimize operating cost, voltage deviation, renewable-energy utilization, battery degradation, control latency, and physical constraint satisfaction.

2. Edge-Cloud Collaborative Digital Twin and Hierarchical Decision-Making Method

2.1 Energy-System Modeling and Edge-Cloud Digital-Twin Architecture

Consider a distribution microgrid with N buses, photovoltaic and wind generation, battery storage, adjustable loads, and a point of common coupling. We define the state as $x_t = \mathrm{col}(V_t, \delta_t, P_t, Q_t, f_t, SOC_t)$ where V_t and δ_t are nodal voltage magnitudes and phase angles, P_t and Q_t are nodal active/reactive injections, f_t is system frequency, and SOC_t is the vector of battery states of charge. The control vector is $u_t = \mathrm{col}(P_t^{ch}, P_t^{dis}, P_t^{dr}, Q_t^{inv}, P_t^{grid})$, The discrete twin model is expressed as

$$x_{t+1} = f(x_t, u_t, d_t; \theta_t) + w_t \tag{1}$$

where x_t, u_t, and d_t denote the state, control, and disturbance vectors at time t, respectively; θ_t denotes the online-calibrated twin parameters; and w_t represents unmodeled dynamics. Multi-source sensor observations satisfy

$$y_t = h(x_t; \theta_t) + v_t \tag{2}$$

The observation function $h(\cdot)$ depends on the measurement type. Direct voltage, frequency, and SOC measurements are selected by the time-varying matrix S_t, while active/reactive power and branch-flow measurements are obtained from the nonlinear AC measurement functions based on v_t, δ_t, and the calibrated admittance matrix $y_t = \mathrm{Y}(\theta_t)$. Thus, $y_t = h(x_t; \theta_t) + v_t, h(x_t; \theta_t) = \mathrm{col}(S_t x_t, h_{PQ}, h_{\mathrm{line}})$, where y_t collects SCADA, smart-meter, and edge-sensor measurements, and v_t denotes measurement noise. Missing measurements are handled by removing the corresponding rows of S_t before synchronization. The active- and reactive-power balances at bus i are

$$P^g_{i,t} + P^{pv}_{i,t} + P^{wt}_{i,t} + P^{dis}_{i,t} - P^{ch}_{i,t} - P^L_{i,t} + P^{dr}_{i,t} = \sum_{j\in\mathcal{N}_i} P_{ij,t} \tag{3}$$

$$Q^g_{i,t} + Q^{inv}_{i,t} - Q^L_{i,t} = \sum_{j\in\mathcal{N}_i} Q_{ij,t} \tag{4}$$

where $P^g_{i,t}$, $P^{pv}_{i,t}$, and $P^{wt}_{i,t}$ denote the outputs of conventional generation, photovoltaic generation, and wind generation, respectively; $P^{ch}_{i,t}$ and $P^{dis}_{i,t}$ denote the battery charging and discharging powers; $P^L_{i,t}$ and $P^{dr}_{i,t}$ denote the load and demand-response terms; $P_{ij,t}$ and $Q_{ij,t}$ denote branch power flows; and $\mathcal{N}_i$ denotes the neighborhood of bus i. The battery-state dynamics and charging/discharging exclusivity constraints are

$$SOC_{t+1} = SOC_t + \frac{\eta_c P^{ch}_t \Delta t}{E_b} - \frac{P^{dis}_t \Delta t}{\eta_d E_b} \tag{5}$$

$$0 \le P^{ch}_t \le z_t P^{max}_{ch}, 0 \le P^{dis}_t \le (1 - z_t) P^{max}_{dis} \tag{6}$$

where η_c and η_d are the charging and discharging efficiencies, E_b is the rated energy capacity, and z_t is the binary charging-state variable. The available photovoltaic and wind powers are given by

$$P^{pv,av}_t = P^r_{pv} \frac{G_t}{G_{STC}} [1 + \kappa_T (T_t - T_{STC})] \tag{7}$$

$$P^{wt,av}_t = \begin{cases} 0, & v_t < v_{ci} \\ P_r \frac{v_t^3 - v_{ci}^3}{v_r^3 - v_{ci}^3}, & v_{ci} \le v_t < v_r \\ P_r, & v_r \le v_t \le v_{co} \\ 0, & v_t > v_{co} \end{cases} \tag{8}$$

where G_t and T_t denote irradiance and module temperature, respectively; κ_T is the temperature coefficient; and v_{ci}, v_r, and v_{co} are the cut-in, rated, and cut-out wind speeds. Adjustable loads satisfy the energy-conservation and comfort constraints

$$\sum_{t\in\mathcal{T}} P^{dr}_t \Delta t = 0, -\rho^- P^L_t \le P^{dr}_t \le \rho^+ P^L_t \tag{9}$$

Figure 1 presents the overall architecture. Physical devices and sensors generate high-frequency data; the edge twin performs time alignment, state estimation, risk assessment, and local fast control; the cloud twin receives event-triggered compressed states and performs global forecasting, rolling optimization, and parameter updating; and the hierarchical decision-maker adjusts the weights of the two control commands according to network and operating conditions. This architecture incorporates the global optimization capability of cooperative MPC for interconnected microgrids [21] while retaining a fast edge-side control loop. Table 1 summarizes the main symbols and variables used in the proposed model.

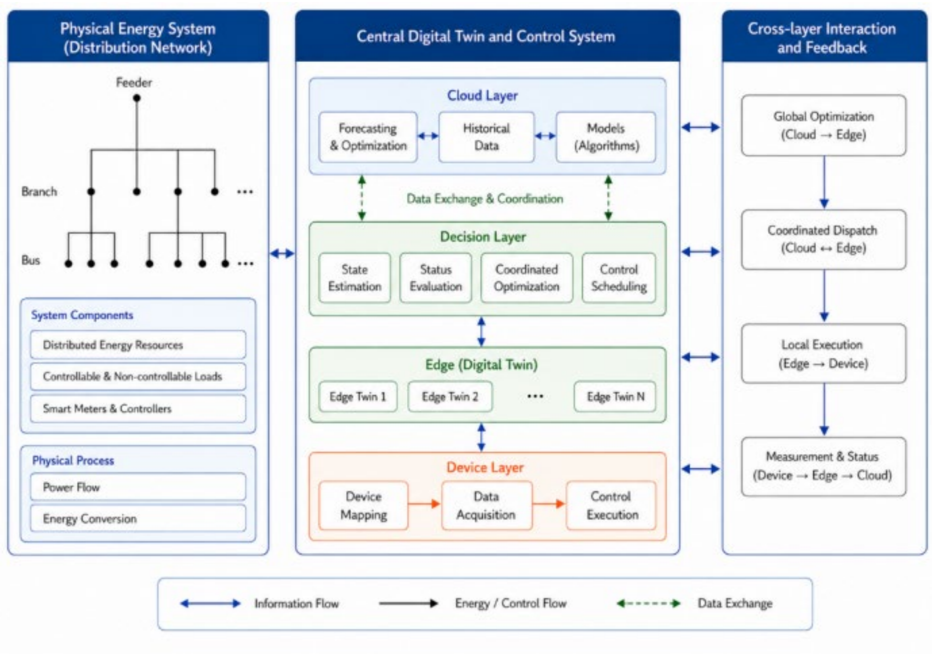

Figure 1. Overall Architecture of the Three-Tier Digital Twin

Table 1. Definitions of Main Symbols and Variables

Symbol	Definition	Symbol	Definition
x_t	System-state vector	u_t	Control vector
d_t	External disturbance	y_t	Multi-source observations
A_t	Energy-network adjacency matrix	h_t	Edge hidden feature
c_t	Estimation confidence	r_t	Operational risk
τ_t	End-to-end control latency	a_t	Cloud-state freshness
u_t^e	Fast edge control	u_t^c	Cloud-optimized control
ω_t	Cloud decision weight	$\mathcal{U}_t$	Time-varying safe feasible set
H	MPC prediction horizon	λ_k	Objective-function weight

2.2 Multi-Source Data Synchronization and Edge State Estimation

Different device sampling periods, network jitter, and clock offsets cause measurements to arrive at edge nodes asynchronously. Let the raw timestamp of the m-th sensor type be $y_{m,k}$. A sliding window of length L is established at edge reference time t, and local linear interpolation yields

$$\tilde{y}_{m,t} = y_{m,k} + \frac{t-t_{m,k}}{t_{m,k+1}-t_{m,k}}(y_{m,k+1} - y_{m,k}) \tag{10}$$

When data are missing or exceed the valid interpolation interval, the mask $m_{m,t}$ and the twin-predicted value are used for compensation

$$y_{m,t}^* = m_{m,t}\tilde{y}_{m,t} + (1 - m_{m,t})\hat{y}_{m,t|t-1} \tag{11}$$

where $m_{m,t} = 1$ indicates a valid observation. Input confidence is defined according to data freshness, missing duration, and residual error as

$$q_{m,t} = \exp(-\alpha_a \Delta a_{m,t} - \alpha_g g_{m,t})\exp[-\alpha_r |y_{m,t}^* - \hat{y}_{m,t|t-1}|] \tag{12}$$

where $\Delta a_{m,t}$ is the observation age, $g_{m,t}$ is the consecutive missing duration, and α_a, α_g, and α_r are decay coefficients. The grid topology is represented by an adjacency matrix incorporating electrical distance

$$A_{ij,t} = \begin{cases} \exp(-|Z_{ij}|/\sigma_z), & (i,j) \in \mathcal{E}_t \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

In the graph attention network (GAT) layer, the unnormalized correlation and normalized weight between buses i and j are respectively defined as

$$e_{ij,t} = \mathrm{LeakyReLU}[a^T(W_h h_{i,t} \parallel W_h h_{j,t} \parallel W_z z_{ij})] \tag{14}$$

$$\alpha_{ij,t} = \frac{\exp(e_{ij,t})}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp(e_{ik,t})} \tag{15}$$

where $h_{j,t}$ denotes the node feature, z_{ij} denotes line-impedance and capacity features, W_h, W_z, and a are learnable parameters, and ‖ denotes concatenation. The graph-encoder output is

$$g_{i,t} = \phi\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \alpha_{ij,t} W_g h_{j,t}\right) \tag{16}$$

To reduce edge inference overhead, the temporal branch adopts depthwise separable causal convolution and a gating unit. The temporal convolution with kernel length K is

$$s_{i,t} = \sum_{k=0}^{K-1} w_k^{dw} \odot g_{i,t-k} \tag{17}$$

$$r_{i,t}^g = \sigma(W_r[s_{i,t}, h_{i,t-1}^T]), h_{i,t}^T = r_{i,t}^g \odot h_{i,t-1}^T + (1 - r_{i,t}^g) \odot \tanh(W_s s_{i,t}) \tag{18}$$

The graph and temporal branches are fused through confidence gating

$$\beta_{i,t} = \sigma(W_b[g_{i,t} \parallel h_{i,t}^T \parallel q_{i,t}] + b_b), f_{i,t} = \beta_{i,t} \odot g_{i,t} + (1 - \beta_{i,t}) \odot h_{i,t}^T \tag{19}$$

The state-estimation and risk heads output

$$\hat{x}_{i,t}^e = W_x f_{i,t} + b_x, c_{i,t} = \exp[-\mathrm{softplus}(W_\sigma f_{i,t} + b_\sigma)] \tag{20}$$

$$r_{i,t} = \sigma[W_r^o(f_{i,t} || \Delta\hat{x}_{i,t}^e \parallel \xi_{i,t}) + b_r^o] \tag{21}$$

where $c_{i,t} \in (0, 1]$ denotes estimation confidence, and $\xi_{i,t}$ denotes threshold-exceedance features for voltage, frequency, SOC, and line loading. The training loss consists of confidence-weighted state error, power-flow residuals, and an uncertainty-calibration term. Figure 2 illustrates the data flow of the edge model.

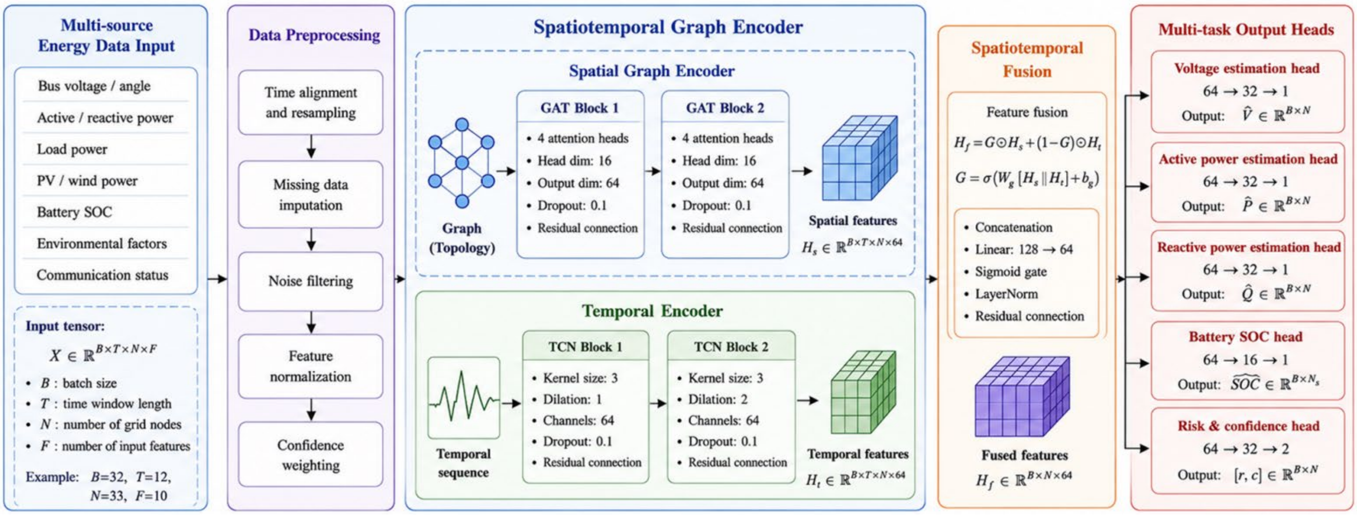

Figure 2. Multi-Source Data Synchronization and Edge State-Estimation Model

The edge model controls model size through parameter sharing and low-rank projection, while only normalization statistics and the final calibration layer are updated online. Graph-neural-network-based state estimation has demonstrated good physical consistency under limited measurements and missing data [22]. In this work, the confidence output is further used directly in the control-weight calculation.

2.3 Cloud Prediction and Multi-Objective Optimal Decision-Making

When the event-trigger condition is satisfied, the k-th edge twin uploads the compressed state $z_{k,t} = C_k[\hat{x}^e_{k,t}, c_{k,t}, r_{k,t}]$. The cloud completes global aggregation according to node coverage and data freshness

$$z_t^c = \sum_{k=1}^{K_e} \pi_{k,t} z_{k,t}, \pi_{k,t} = \frac{c_{k,t}\exp(-\mu a_{k,t})}{\sum_l c_{l,t}\exp(-\mu a_{l,t})} \tag{22}$$

The cloud predictor takes the global graph state, weather, electricity price, and historical control as inputs and recursively generates H-step load, renewable-energy, and state trajectories

$$\hat{d}_{t+h|t} = F_\psi^{(h)}(z_t^c, d_{t-L+1:t}, u_{t-L+1:t-1}), h = 1, \dots, H \tag{23}$$

The prediction loss consists of a multi-step Huber error and a physical-boundary penalty

$$\mathcal{L}_{pred} = \sum_{h=1}^{H} \gamma_h \text{Huber}(d_{t+h} - \hat{d}_{t+h|t}) + \lambda_p \| [g(\hat{d}_{t+h|t})]_+ \|_2^2 \tag{24}$$

At each cloud control period, the rolling MPC problem is solved

$$J_t = \sum_{h=0}^{H-1} (\lambda_c C_{t+h}^{grid} + \lambda_b C_{t+h}^{deg} + \lambda_v C_{t+h}^{volt} + \lambda_r C_{t+h}^{curt} + \lambda_\tau C_{t+h}^{lat}) + \lambda_T V_f(x_{t+H}) \tag{25}$$

where the individual cost terms are defined as

$$C_t^{grid} = p_t^{buy}[P_t^{grid}]_+\Delta t - p_t^{sell}[-P_t^{grid}]_+\Delta t \tag{26}$$

$$C_t^{deg} = c_b[(P_t^{ch} + P_t^{dis})\Delta t] + c_s(SOC_t - SOC_{t-1})^2 \tag{27}$$

$$C_t^{volt} = \sum_i (V_{i,t} - V_{ref})^2 + \rho_v([V_i^{min} - V_{i,t}]_+^2 + [V_{i,t} - V_i^{max}]_+^2) \tag{28}$$

$$C_t^{curt} = c_{curt}(P_t^{pv,av} - P_t^{pv} + P_t^{wt,av} - P_t^{wt}) \tag{29}$$

The latency cost C_t^{lat} constrains cloud control to finish before its result becomes stale, while V_f is the terminal penalty on battery energy and power exchange. The optimization must also satisfy Eqs. (3)-(9), voltage limits, line-capacity limits, and ramp-rate constraints. MPC can incorporate economic objectives, storage dynamics, and hard constraints within a unified rolling-optimization framework, and general design methodologies have demonstrated its transferability across different microgrid architectures [23]. Experimental studies on multiple-dynamic-matrix MPC have also shown that hierarchical storage control can improve DC microgrid dynamics [24]. Figure 3 presents the cloud prediction-optimization procedure.

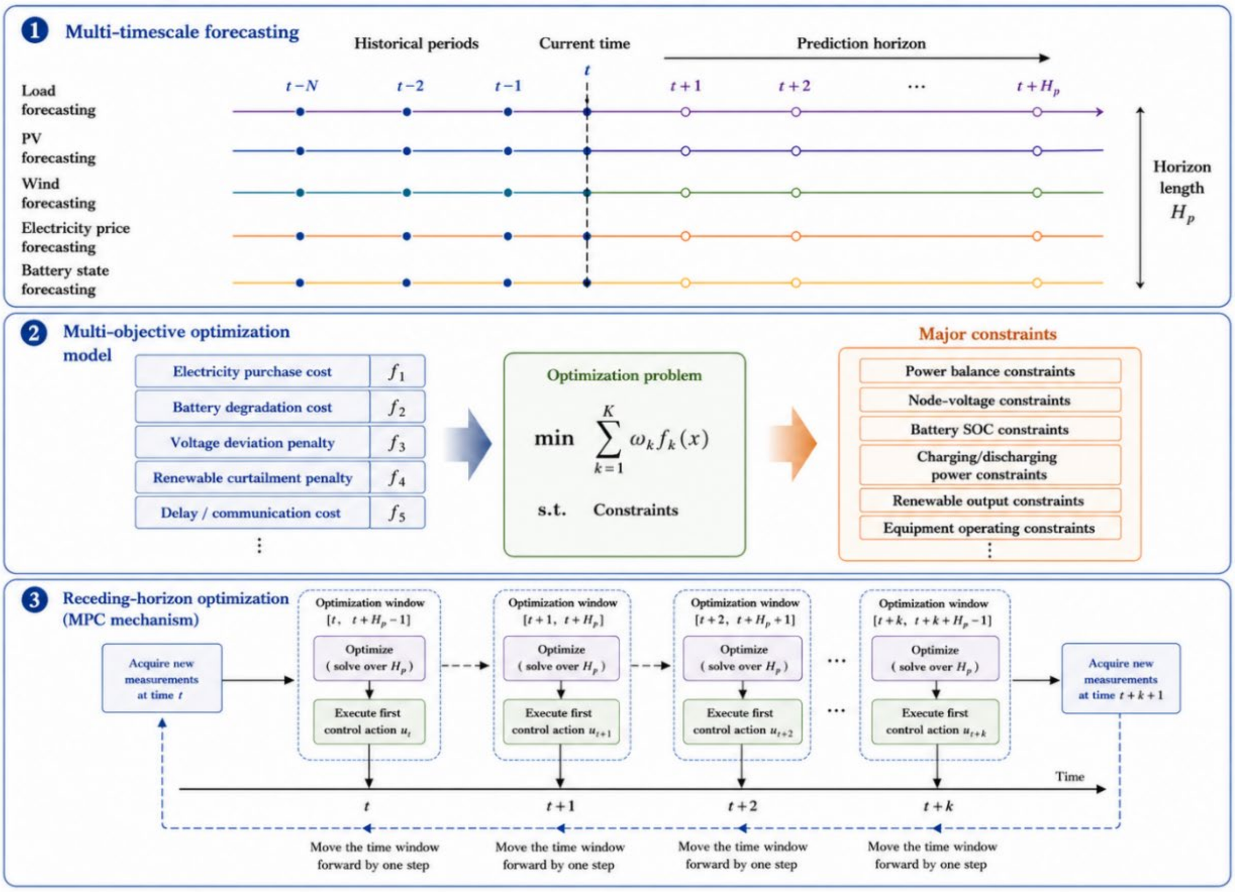

Figure 3. Cloud-Based Global Prediction and Multi-Objective Optimization Model

2.4 Latency-Aware Hierarchical Decision-Making and Safe Control

The end-to-end control latency consists of device sampling, uplink communication, cloud queuing and optimization, downlink communication, edge fusion, and actuation

$$\tau_t = \tau_t^{sen} + \tau_t^{up} + \tau_t^{queue} + \tau_t^{cloud} + \tau_t^{down} + \tau_t^{edge} + \tau_t^{act} \quad (30)$$

Cloud-state freshness is defined using the age of information

$$a_t = t - t_{gen}^c, a_t \leftarrow 0 \text{after a valid cloud update} \quad (31)$$

Let u_t^e denote the fast edge control and u_t^c denote the cloud-optimized control [25]. The cloud weight is jointly determined by latency, state age, edge confidence, risk, and computational load

$$\omega_t = \sigma[\theta_0 - \theta_\tau \bar{\tau}_t - \theta_a \bar{a}_t + \theta_c \bar{c}_t - \theta_r \bar{r}_t - \theta_l \ell_t] \quad (32)$$

For each normalized factor $q_j, \partial\omega_t / \partial q_j = s_j \theta_j \omega_t (1 - \omega_t)$, where $s_j = -1$ for latency, information age, risk, and computational load, and $s_j = +1$ for confidence. Since ω_t $(1-\omega_t) \leq 1/4$, the influence of each factor is bounded by $\theta_j/4$, ensuring smooth weight variation and reducing switching chattering. A learned policy is not adopted because the monotone logistic form provides clearer interpretability, predictable factor influence, and negligible edge-side computation. Coefficient sensitivity is evaluated by perturbing each θ_j around its validated value while keeping the others fixed.

In contrast to a static fusion coefficient or discrete priority-based switching, Eq. (32) maps the five real-time indicators to a continuous cloud-authority weight, allowing control responsibility to shift smoothly before a hard mode transition is required. This joint five-factor authority-allocation logic, rather than the individual GAT, MPC, or QP components, constitutes the main algorithmic novelty of EC-HDT.

$$\tilde{u}_t = (1 - \omega_t) u_t^e + \omega_t u_t^c + K_\Delta (\hat{x}_t^c - \hat{x}_t^e) \quad (33)$$

where K_Δ is the twin-state deviation compensation matrix. An upload is triggered when the local state change, risk, or information age reaches its threshold

$$\delta_t = \mathbb{I}(\| \hat{x}_t^e - \hat{x}_{t_k}^e \|_{W_x} > \varepsilon_x \vee r_t > \varepsilon_r \vee a_t > \varepsilon_a) \quad (34)$$

The final control is projected onto the feasible set through a quadratic-programming safety layer

$$u_t^* = \arg\min_{u \in \mathcal{U}_t} \frac{1}{2} \| u - \tilde{u}_t \|_{R_s}^2 \quad (35)$$

where $\mathcal{U}_t$ is defined by the power-balance, SOC, voltage, line-capacity, charging/discharging exclusivity, and ramp-rate constraints. The consistency loss between the edge and cloud twins is

$\mathcal{L}_{con} = \| M_t \odot (\hat{x}_t^e - \hat{x}_t^c) \|_1 + \chi \| \theta_t^e - \Pi(\theta_t^c) \|_2^2$
(36)

The overall training objective is written as $L = \lambda_x \mathcal{L}_{state} + \lambda_f \mathcal{L}_{flow} + \lambda_u \mathcal{L}_{unc} + \lambda_p \mathcal{L}_{pred} + \lambda_c \mathcal{L}_{con} + \lambda_s \mathcal{L}_{safe}$
(37)

Only the first control action of the rolling sequence is applied in each cloud period, and the twin parameters are updated using new observations

$$u_t \leftarrow u_{t|t}^*, \theta_{t+1} = \theta_t - \eta_\theta \nabla_\theta \ell(y_{t+1}, \hat{y}_{t+1|t}) \quad (38)$$

Figure 4 illustrates the closed-loop decision process. When the network is stable, cloud results are fresh, and edge-estimation confidence is high, ω_t increases to exploit global economic benefits. When communication latency, state age, or operational risk increases, control authority shifts toward the edge. Regardless of the weight variation, the safety layer ensures that the final action satisfies the physical constraints. Recent techno-economic MPC studies have shown that real-time prediction and multi-objective constraints can substantially improve microgrid economics. The distinction of this work is that tail latency and decision-mode switching are explicitly incorporated into the closed loop.

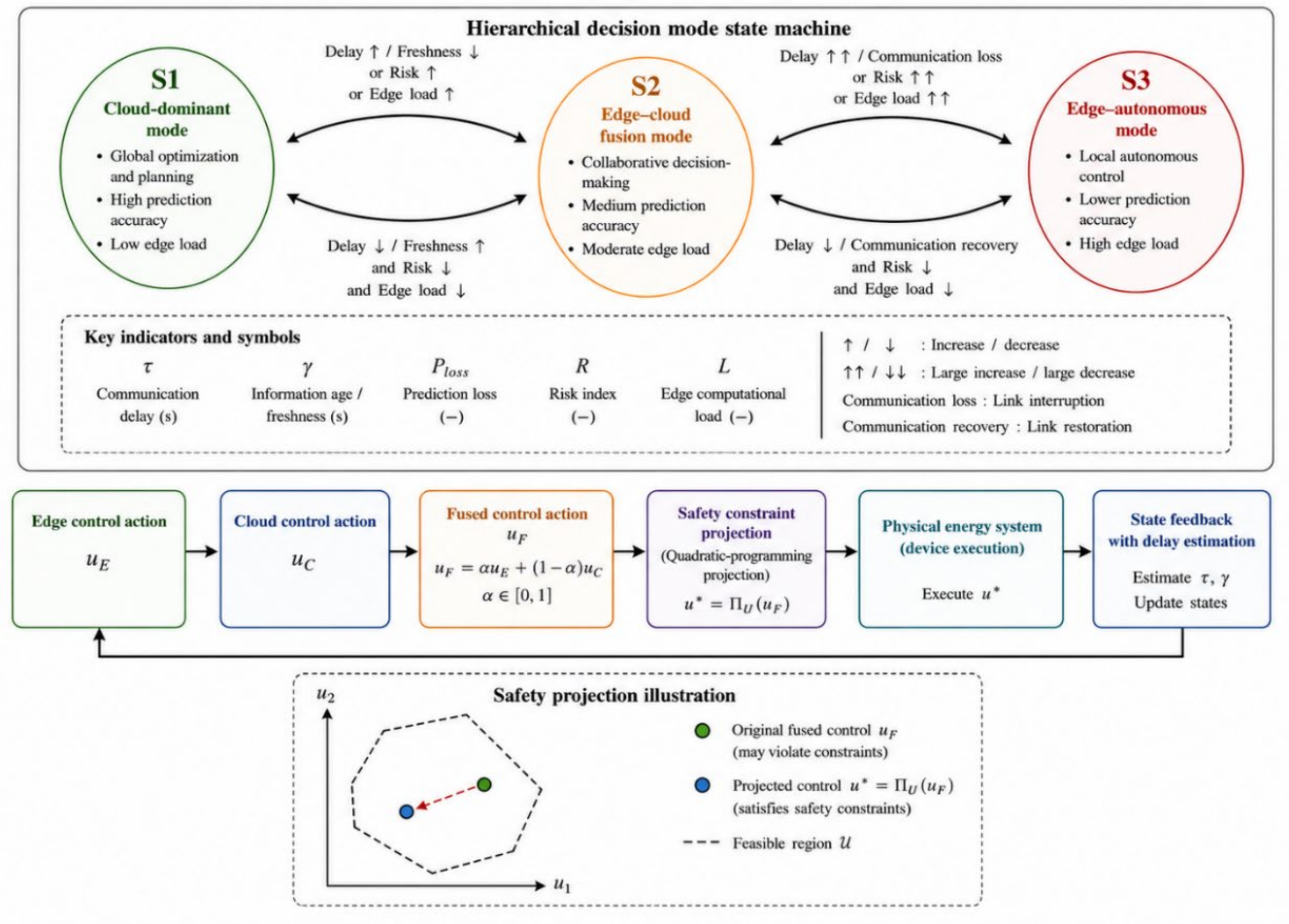

Figure 4. Latency-Aware Hierarchical Decision-Making and Safe Closed-Loop Control

3 Dataset and Experimental Setup

3.1 Data Sources and Simulation System

The experiments use the IEEE 33-bus radial distribution system in an Open Distribution System Simulator (OpenDSS)–Python co-simulation environment. The base active load is scaled to 3.72 MW and the reactive load is 2.30 Mvar. Photovoltaic systems with a total capacity of 2.4 MW are connected at buses 6, 14, and 30; a 0.8 MW wind turbine is connected at bus 18; battery energy storage systems with a total rated power of 1.5 MW and total capacity of 4.2 MWh are installed at buses 8, 22, and 31; and transferable loads are placed at buses 9, 16, 25, and 32. The load profiles are formed by normalized combinations of public residential electricity-consumption data, while photovoltaic and meteorological sequences are generated from public solar-energy data at a 5 min resolution. The data are divided into training, validation, and test sets in proportions of 70%, 15%, and 15%, respectively. The test set includes normal days, load disturbances, rapid photovoltaic drops, missing data, sensor noise, communication congestion, packet loss, and cloud-outage scenarios. The main dataset characteristics, system parameters, and experimental settings are summarized in Table 2.

Random scenarios are generated using Monte Carlo simulation. Load-forecast errors follow a truncated normal distribution, photovoltaic errors use a skewed distribution to simulate cloud disturbances, and communication round-trip delay is modeled by a mixture of a lognormal body and a congestion tail. Studies on randomized multi-objective microgrid scheduling with mobile storage indicate that uncertainties in load, renewable generation, and storage location should be considered jointly [26]. Data-driven microgrid control studies also show that scenario-based optimization improves regulation stability under distributed-energy-resource variability [27].

Table 2. Dataset and Energy-System Configuration

Item	Configuration	Item	Configuration
Distribution system	IEEE 33-bus	Simulation platform	OpenDSS 0.9 + Python 3.11
Time resolution	5 min; control period: 1 s	Sample size	105,120 time steps
Data split	70%/15%/15 %	Training window	12 steps; prediction H = 12
PV capacity	2.4 MW	Wind capacity	0.8 MW
Storage configuration	1.5 MW/4.2 MWh	SOC range	0.15-0.90
Communication delay	20-300 ms	Packet-loss rate	0-20%
Noise level	0-5%	Missing-data rate	0-40%
Repeated experiments	10 random seeds	Significance test	Paired t-test, $p < 0.05$

3.2 Comparative Methods

Eight methods are compared. Cloud-MPC is a purely cloud-based centralized MPC method using weighted least squares (WLS) for state estimation; Edge-DDC is an edge-based droop and decentralized control method; DT-MPC is a centralized digital-twin-driven MPC method; LSTM-EMS is a long short-term memory (LSTM)-based energy management system (EMS) that performs cloud scheduling using LSTM forecasts; GNN-EMS is a GNN-based EMS that uses graph attention network (GAT)-based state estimation followed by single-layer control; EC-Fixed is an edge–cloud model with a fixed fusion weight of 0.5; EC-NoSP is the edge–cloud model without safety projection; and EC-HDT is the complete model. All learning-based models use the same training set, prediction horizon, and similar hidden dimensions. The MPC methods use identical device models, constraints, and electricity-price profiles. Multilevel microgrid energy-management research indicates that upper- and lower-level objectives and uncertainty treatments can substantially affect cost and stability [28]. Therefore, this study keeps the physical resources and objective weights unchanged and varies only the state-estimation and control architectures. The configurations of all compared methods are summarized in Table 3.

Table 3. Comparative Models and Experimental Parameters

Method	State model	Control strategy	Deployment	Parameters/M	Latency awareness
Cloud-MPC	WLS + prediction	Centralized MPC	Cloud	-	No
Edge-DDC	Local measurements	Decentralized rules	Edge	-	No
DT-MPC	Centralized DT	Rolling MPC	Cloud	1.84	No
LSTM-EMS	LSTM	Predictive scheduling	Edge-cloud	1.26	No
GNN-EMS	GAT	Single-layer optimization	Edge-cloud	1.41	No
EC-Fixed	GAT-TCN	Fixed fusion	Edge-cloud	1.58	Fixed weight
EC-NoSP	GAT-TCN	Dynamic fusion	Edge-cloud	1.61	Yes
EC-HDT	GAT-TCN	Dynamic fusion + QP	Edge-cloud	1.64	Yes

3.3 Evaluation Metrics

State estimation is evaluated using MAE, RMSE, and MAPE, defined as

$$MAE = \frac{1}{n}\sum_{k=1}^{n} |x_k - \hat{x}_k| \tag{39}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{x}_k)^2} \tag{40}$$

where n is the number of test samples and ε prevents the denominator from approaching zero. The control success rate is defined as the proportion of time steps in which control is completed within the deadline without violations of the hard voltage, frequency, or SOC constraints

$$R_{succ} = \frac{N_{deadline \cap safe}}{N_{total}} \times 100\% \tag{41}$$

Daily operating cost, renewable-energy utilization, average voltage deviation, mean and 95th-percentile end-to-end latency, communication traffic, SOC violation rate, and recovery time are also reported. Stochastic-optimization studies show that a single economic metric cannot fully reflect trade-offs among energy storage, emissions, and renewable-energy hosting capacity [29]. Reviews of 100% renewable microgrids likewise emphasize the need to jointly evaluate economics, stability, and flexibility [30].

3.4 Implementation Details

The model is trained using PyTorch 2.4. The graph-attention branch contains two layers with four heads per layer and a hidden dimension of 64. The temporal branch uses three depthwise separable convolutional layers with kernel sizes of 3, 5, and 7, followed by a 64-dimensional gating unit. AdamW is used with an initial learning rate of 1×10^{-3}, weight decay of 1×10^{-4}, batch size of 64, and a maximum of 150 epochs; early stopping is applied if the validation score does not improve for 20 epochs. After structured pruning and INT8 quantization, the edge model is deployed on a four-core ARM edge node, while the cloud uses an eight-core CPU and a mid-range GPU. The reported latency is obtained from the OpenDSS–Python software-in-the-loop environment rather than HIL or field measurements. Practical deployment may introduce additional delays from communication interfaces, protocol processing, clock drift, operating-system scheduling, and actuator dynamics. The edge-control period is 1 s and the cloud-optimization period is 60 s. The event thresholds are $\varepsilon_x = 0.035$, $\varepsilon_r = 0.65$, and $\varepsilon_a = 2$ s.

An additional IEEE 69-bus and IEEE 118-bus case is included to evaluate adaptability to topological unobservability and system-scale expansion. Topology-aware GNN studies show that explicitly encoding line structure can improve state-estimation robustness in PMU-unobservable systems [31]. Real-time multi-stability assessment studies further demonstrate that graph models are suitable for rapid risk screening [32].

4 Experimental Results and Analysis

4.1 Overall Performance Comparison

To comprehensively evaluate EC-HDT in terms of state awareness, operating economics, control safety, and real-time response, it is compared with centralized cloud control, purely edge control, conventional digital-twin control, LSTM-based predictive control, GNN-based state estimation, and two ablation models. The evaluation metrics include state-estimation error, daily operating cost, voltage deviation, renewable-energy utilization, control success rate, and end-to-end latency, allowing the global optimization capability, local response speed, and safety-constraint satisfaction of different computing deployments and decision mechanisms to be analyzed.

Figure 5 compares the end-to-end control latency of different methods. Cloud-MPC and DT-MPC show mean latencies of 247.3 ms and 218.2 ms, with P95 values of 418.6 ms and 361.9 ms, respectively. Edge-DDC achieves the lowest mean latency of 31.2 ms but lacks cloud-based global optimization. EC-HDT attains a mean latency of 56.4 ms and a P95 latency of 89.4 ms, reducing them by 31.88% and 37.04% relative to EC-Fixed. Compared with EC-NoSP, the additional 5.7 ms latency introduced by safety projection improves constraint satisfaction. Relative to the 1 s control period, the mean and P95 latencies occupy only 5.64% and 8.94% of the cycle, leaving substantial timing headroom for hardware variability, communication jitter, and actuation overhead. Recent 2025 studies have also investigated digital-twin-enabled cloud-edge coordination and three-layer edge-cloud microgrid architectures, reporting improvements in latency, resource utilization, scalability, and real-time monitoring [33,34].

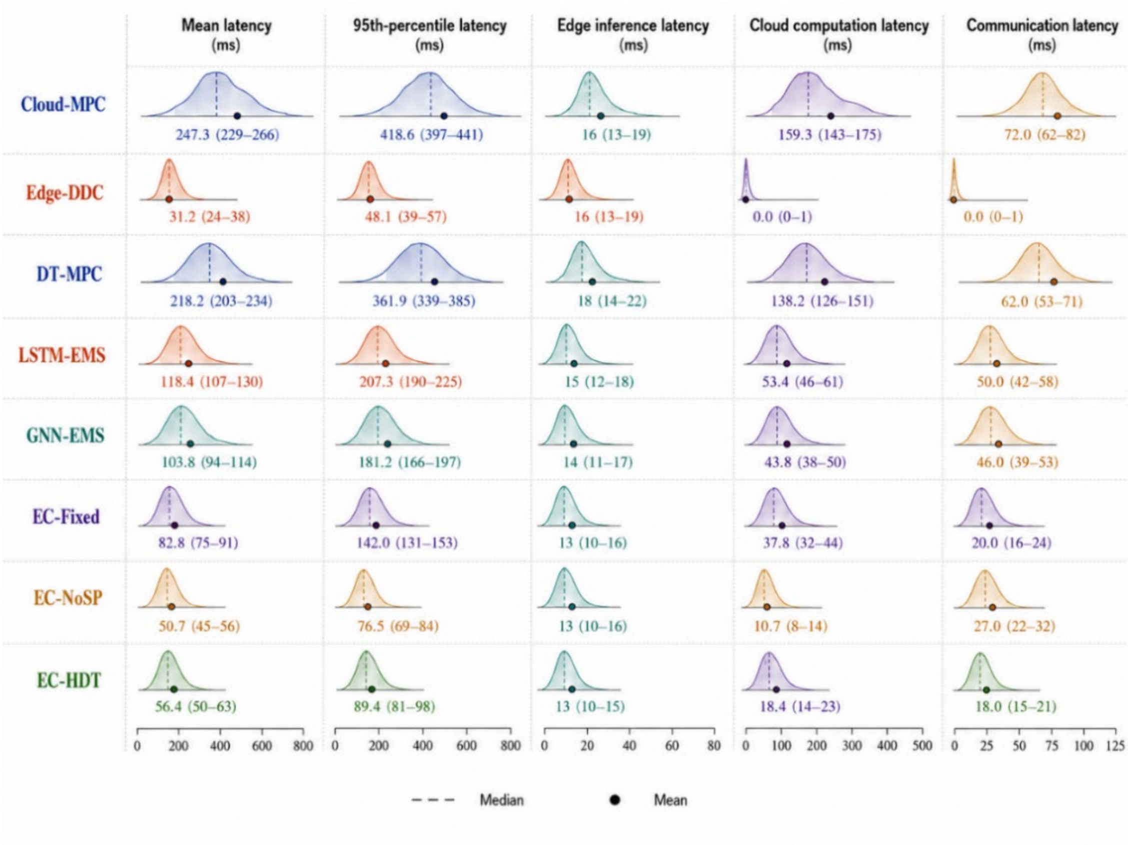

Figure 5. Comparison of End-to-End Control Latency Across Methods

Figure 6 shows the dynamic regulation process of EC-HDT under a sudden load increase. After approximately 40 min, the load rises from about 2.2 MW to 3.1 MW. The battery rapidly increases its discharge power, while grid-import power jointly compensates for the resulting power deficit. During the disturbance, the minimum bus voltage drops to approximately 0.97 p.u. and the minimum frequency deviation reaches approximately -0.11 Hz before gradually recovering. Regulation is provided mainly by the battery and the main grid, with flexible loads participating in peak shaving. These results show that the proposed method can rapidly restore power balance and suppress voltage and frequency fluctuations under abrupt load changes.

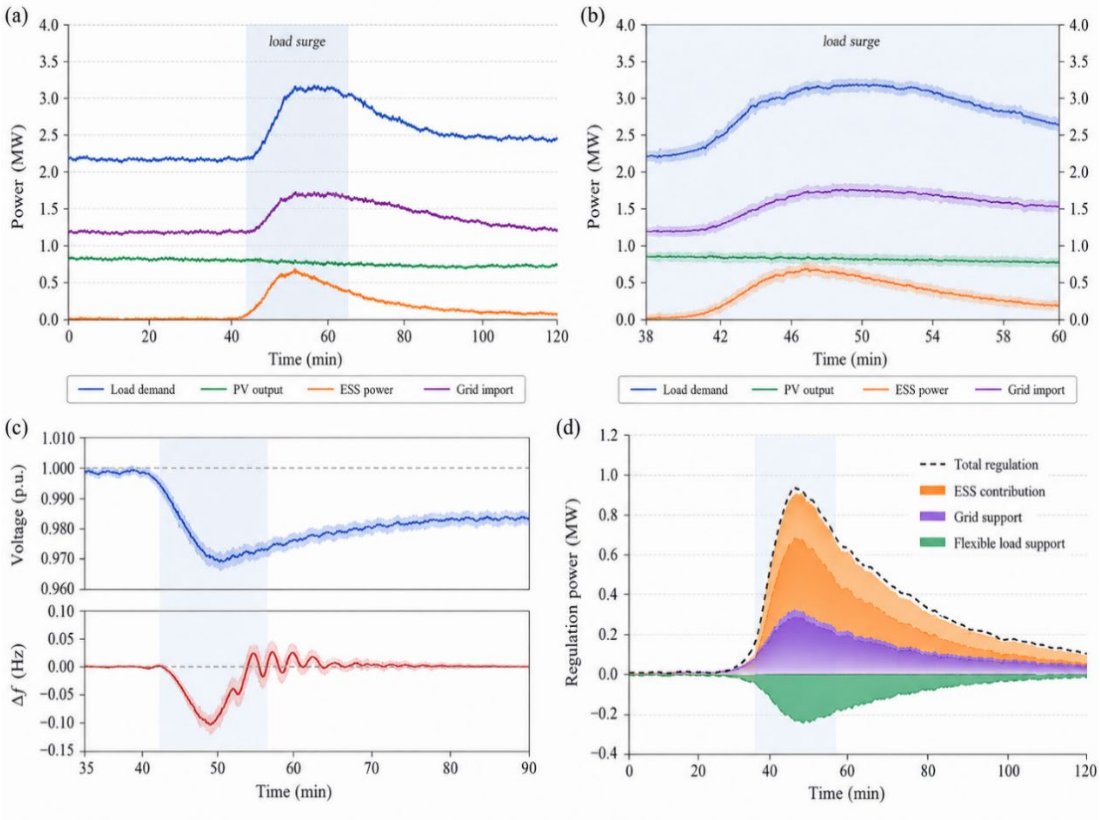

Figure 6. Dynamic Regulation Under a Sudden Load Change

Figure 7 compares the state-estimation error distributions of different methods. For EC-HDT, the median absolute errors of voltage, power, frequency, and SOC estimation are 0.0068 p.u., 0.112 MW, 0.062 Hz, and 1.25%, respectively, all lower than those of the other methods. Relative to EC-HDT without latency awareness, the four errors are reduced by 12.82%, 10.40%, 11.43%, and 19.35%, respectively, indicating that the latency-aware mechanism effectively improves state-estimation accuracy and stability.

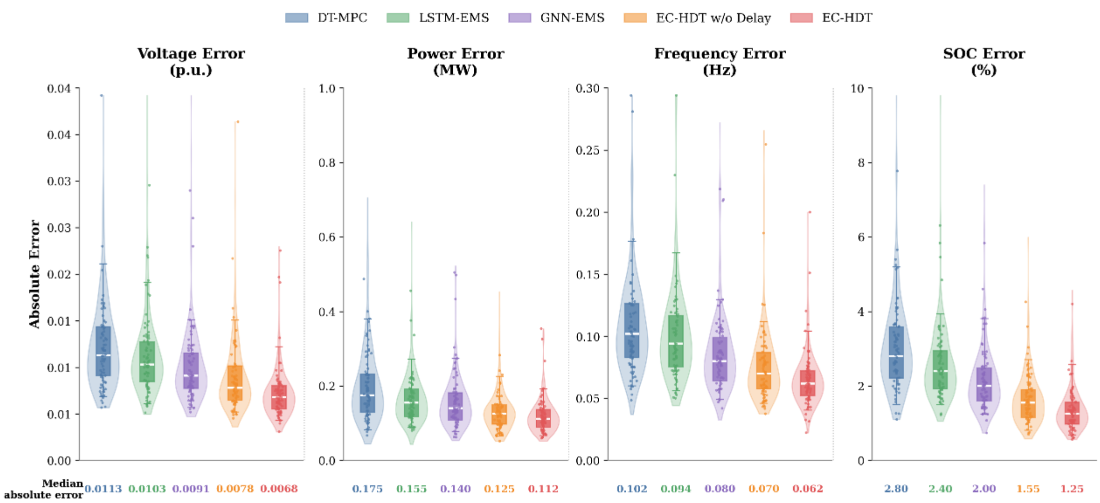

Figure 7. State-Estimation Error Distributions of Different Methods

Figure 8 compares the methods across six dimensions: estimation accuracy, prediction accuracy, stability, economy, renewable-energy utilization, and low-latency performance [35]. EC-HDT exhibits balanced performance across all dimensions and achieves an overall score of 0.93, outperforming EC-HDT without latency awareness, Cloud-MPC, GNN-Estimator, and Edge-Only. Cloud-MPC performs well in economy and renewable-energy utilization but poorly in low-latency performance. Edge-Only provides strong real-time performance but weaker estimation accuracy, stability, and economy. These results indicate that EC-HDT achieves a better balance between global optimization and rapid response.

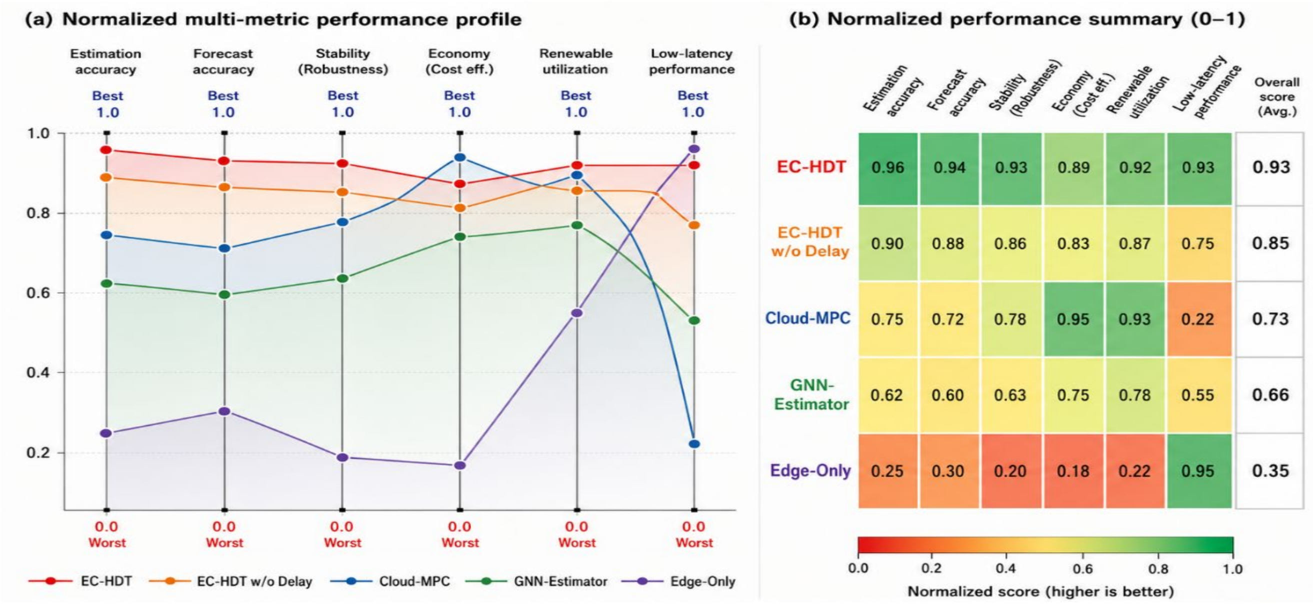

Figure 8. Normalized Overall Performance Comparison of Different Methods

4.2 Robustness and Engineering Applicability

Table 4 shows that under 5% measurement noise, 20% missing data, and 15% packet loss, the control success rates of EC-HDT are 98.9%, 98.1%, and 97.8%, respectively. During a 60 s cloud outage, the dynamic weight rapidly shifts toward the edge, reducing the mean latency to 42.7 ms while maintaining a success rate of 96.5%, at the cost of an MAE increase to 0.0102 p.u. In the IEEE 118-bus extension, the mean latency is 73.4 ms, indicating that the model remains below 100 ms as system scale increases. This result provides preliminary evidence of computational scalability within the simulation environment rather than validation of practical large-scale deployment. Spatiotemporal graph models for short-term load forecasting can effectively capture regional coupling [36], while physics-guided probabilistic power-flow studies show that topology priors improve consistency under stochastic scenarios [37]. Together, these findings support the modeling choices adopted by EC-HDT for scale expansion.

Table 4. Robustness Results Under Different Abnormal Scenarios

Scenario	Success rate/%	MAE/p.u.	Mean latency/ms	SOC violation rate/%
Normal operation	99.6	0.0069	52.1	0.0
5% measurement noise	98.9	0.0082	55.8	0.0
20% missing data	98.1	0.0094	58.9	0.1
15% packet loss	97.8	0.0089	64.3	0.1
60 s cloud outage	96.5	0.0102	42.7	0.3
25% load step	97.2	0.0098	61.5	0.2
IEEE 118-bus	96.8	0.0105	73.4	0.2

Figure 9 illustrates the robustness of EC-HDT under communication abnormalities. Figure 9(a) shows that the control success rate generally remains above 90% under low-to-moderate communication delays. When the delay increases to 160 ms, the success rate declines to 78%-90%. In Fig. 9(b), predicted uncertainty is strongly positively correlated with the actual control error, with a Pearson correlation coefficient of 0.87, indicating that the model output uncertainty effectively reflects control risk under different disturbance intensities.

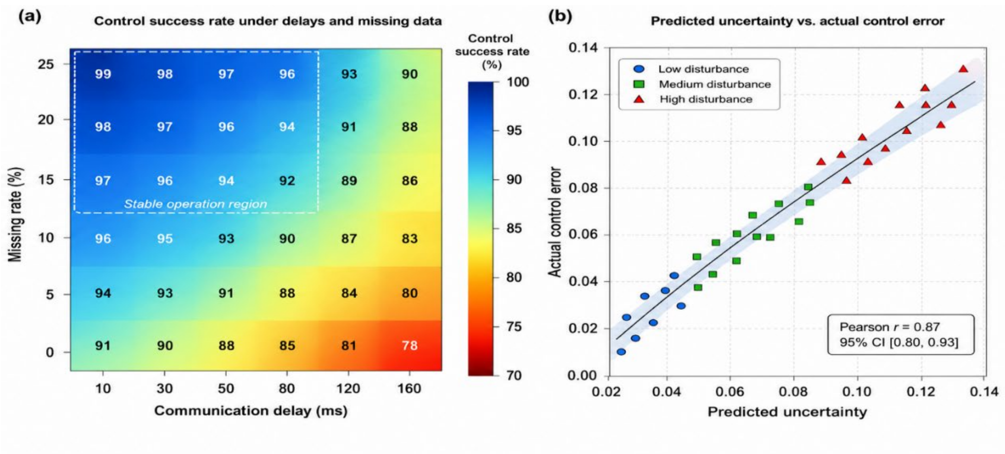

Figure 9. Control Robustness and Uncertainty Calibration Under Communication Delay and Missing Data

4.3 Practical Deployment and Scalability Considerations

The present results are obtained from an OpenDSS–Python software-in-the-loop environment and therefore do not fully capture communication hardware, clock drift, protocol processing, converter/protection dynamics, or actuator delays. Thus, the reported 56.4 ms mean latency and 99.2% control success rate should be interpreted as simulation-level results rather than hardware guarantees. For model maintenance, Eq. (38) only updates bounded calibration variables while keeping the GAT-TCN backbone frozen; unsuccessful updates are rejected. At cold start, the controller relies on current measurements, the physics-based twin estimate, and the QP safety layer until sufficient historical data are available.

To further examine computational scalability, the IEEE 118-bus test is supplemented with edge inference time, memory consumption, and communication traffic, as summarized in Table 5.

Table 5. Computational scaling under different network sizes

System	Peak edge memory	Edge inference time	Traffic	Mean latency
IEEE 33 bus	132 MB	11.8 ms	412 MB/day	56.4 ms
IEEE 69-bus	154 MB	17.6 ms	523 MB/day	64.9 ms
IEEE 118-bus	186 MB	24.7 ms	685 MB/day	73.4 ms

Memory, inference time, and communication traffic increase with network size, while the mean latency remains below the 1 s control deadline. These results indicate computational adaptability within the tested simulation environment, rather than validated field scalability.

Cybersecurity and interoperability remain open issues. Since latency and information freshness affect the cloud weight in Eq. (32), manipulated timestamps or delay measurements may bias authority allocation. Practical deployment should therefore include timestamp validation, abnormal-delay detection, authenticated communication, and conservative fallback. EC-HDT can interface with SCADA/EMS/DMS through protocol adapters such as IEC 61850, DNP3, MQTT, or OPC UA, but HIL, multi-vendor interoperability, and field cybersecurity tests remain future work.

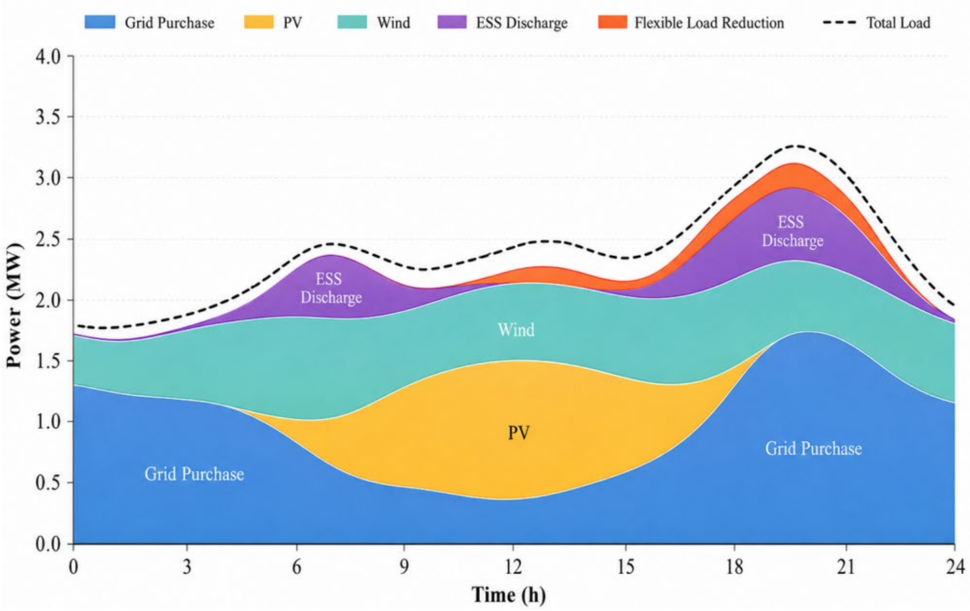

Figure 10. Energy-Scheduling Composition Over a Typical Day

4.4 Dynamic Regulation, Ablation, and Sensitivity Analysis

Figure 10 presents the coordinated scheduling of different energy resources over a typical day. Increased photovoltaic and wind generation during daytime reduces grid electricity purchases. During the morning and evening load peaks, battery discharge and flexible-load curtailment jointly provide regulation [38]. The results show that EC-HDT dynamically coordinates grid purchases, energy storage, and demand response according to variations in load and renewable generation, thereby achieving peak shaving, valley filling, and priority utilization of renewable energy.

Table 6 shows that removing edge state estimation increases the MAE by 19.74% and the mean latency to 72.6 ms, indicating that replacing local synchronization with cloud states degrades both accuracy and real-time performance. Removing cloud prediction increases cost by 5.77% but slightly reduces mean latency, showing that global prediction mainly contributes to economic performance [39,40]. Without latency-aware weighting, the mean latency rises to 79.9 ms and the control success rate falls to 96.2%. Removing event-triggered communication slightly improves estimation accuracy, but communication traffic increases from 412 MB/day to 612 MB/day, an increase of 48.54%. Removing the safety projection reduces cost and latency but lowers the control success rate by 4.9 percentage points.

Removing cloud prediction reduces mean latency by only 1.6 ms (56.4 to 54.8 ms; 0.16% of the 1 s control period) but increases daily cost by CNY 67.3 (5.77%) and lowers control success from 99.2% to 97.1%. Thus, cloud prediction adds minimal latency while delivering clear economic and reliability benefits. The edge-only mode remains suitable when economic dispatch is unnecessary, whereas normal grid-connected operation favors retaining cloud prediction.

Table 6. Ablation Results

Configuration	MAE/p.u.	Cost/CNY	Mean latency/ms	Traffic/MB day^-1	Success rate/%
Complete EC-HDT	0.0076	1166.2	56.4	412	99.2
Without edge estimation	0.0091	1204.8	72.6	458	96.7
Without cloud prediction	0.0088	1233.5	54.8	365	97.1
Without latency awareness	0.0085	1195.4	79.9	438	96.2
Without event triggering	0.0075	1164.7	57.1	612	99.3
Without safety projection	0.0080	1149.5	50.7	408	94.3
Fixed fusion weight	0.0087	1208.6	82.8	436	98.4

Figure 11 shows the trade-off between operating cost and mean control latency for different configurations. The complete EC-HDT achieves a mean latency of 56.4 ms and a daily operating cost of CNY 1,166.2, placing it in the low-cost, low-latency region of the feasible Pareto frontier. Although removing the safety projection further reduces cost and latency, its control success rate falls below 95%, making it infeasible from a safety perspective. The higher latency and cost of the no-latency-awareness and fixed-fusion-weight configurations demonstrate that dynamic decision weighting helps balance real-time performance and economic efficiency.

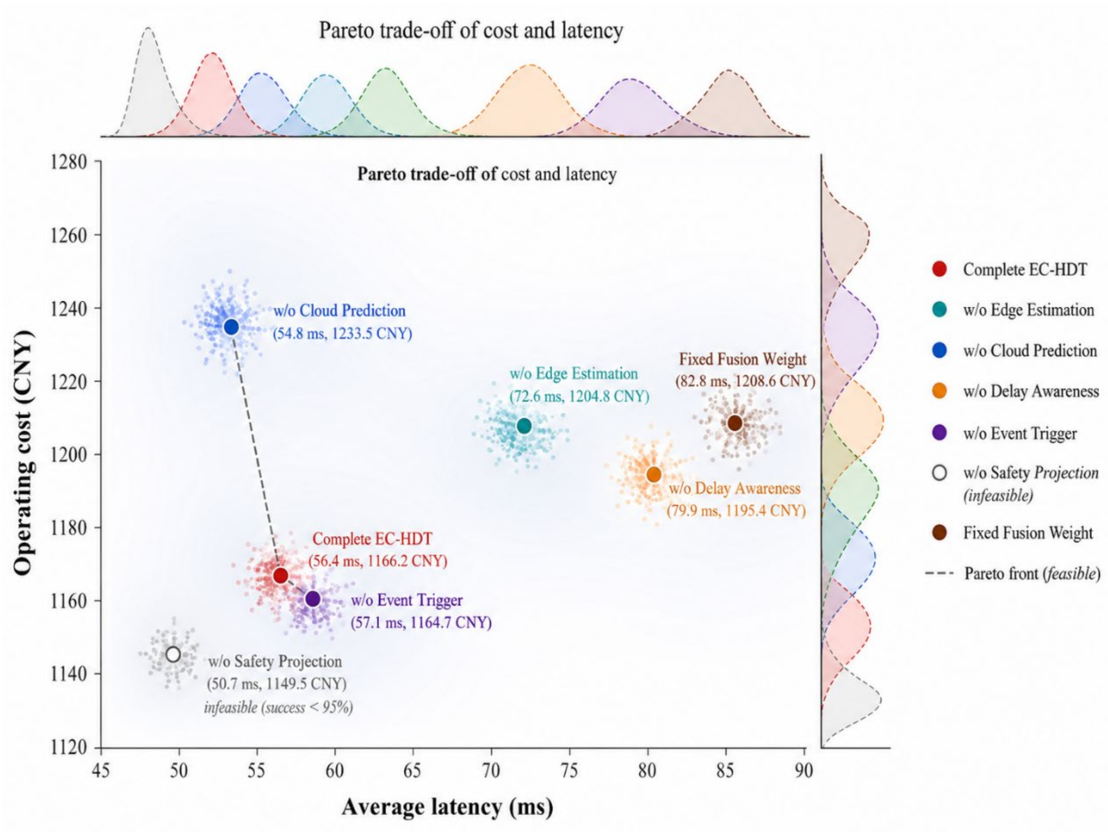

Figure 11. Pareto Trade-Off Between Operating Cost and Control Latency

Figure 12 shows that the median control response times under normal operation, load disturbances, and photovoltaic fluctuations are approximately 35 ms, 51 ms, and 43 ms, respectively. Under communication congestion, the median increases to approximately 67 ms and the distribution is the widest, indicating that network queuing and transmission variability substantially increase tail latency. After a link outage, the system switches to edge-autonomous mode and the median response time decreases to approximately 59 ms; however, state-estimation accuracy and operating economics deteriorate because global cloud information is unavailable. Related studies show that demand response and uncertainty modeling can reduce microgrid operating risk [41], while fast power allocation in hybrid-storage MPC can improve voltage regulation and power quality [42].

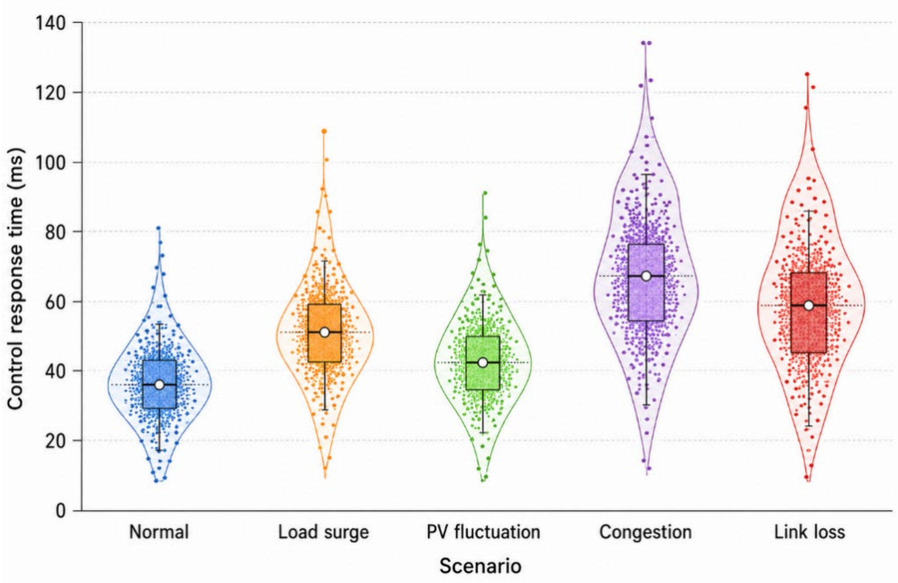

Figure 12. Control-Response-Time Distributions Under Different Scenarios

5. Conclusion

This study addresses the coordinated regulation of low latency, safety, and economy under conditions of high penetration of distributed energy resources by proposing an edge–cloud collaborative hierarchical digital twin model, EC-HDT. Through rapid edge-side perception, cloud-based rolling optimization, dynamic weight allocation, and safety projection, the model achieves coordinated optimization of state estimation, control latency, operating cost, and safety constraints.

(1) Experimental results show that EC-HDT achieves a node-voltage estimation MAE of 0.0076p.u. The median absolute errors for voltage, power, frequency, and SOC estimation are 0.0068p.u., 0.112 MW, 0.062 Hz, and 1.25%, respectively. Compared with the model without delay awareness, the four errors are reduced by 12.82%, 10.40%, 11.43%, and 19.35%, respectively. These results indicate that the proposed data synchronization and delay-aware mechanisms can effectively reduce estimation deviations caused by asynchronous measurements and improve the accuracy and stability of multi-source state perception.

(2) In terms of real-time performance and economic efficiency, the average control latency and the 95th-percentile control latency of EC-HDT are 56.4ms and 89.4ms, respectively, representing reductions of 31.88% and 37.04% compared with the fixed-weight model. The daily operating cost is CNY 1,166.2, which is 3.51% lower than that of the fixed-fusion-weight scheme, while the renewable-energy utilization rate increases by 2.1 percentage points. These results demonstrate that the dynamic edge–cloud decision mechanism can retain the global optimization advantage of the cloud while exploiting the rapid response capability of the edge, thereby coordinating control performance and operating economy.

(3) The complete model achieves a 99.2% control success rate, 4.9 percentage points higher than the model without safety projection and remains above 96.5% under simulated abnormal conditions. In the IEEE 118-bus case, the mean latency is 73.4 ms, indicating computational scalability within the tested simulation environment. However, HIL or field validation is still required to assess practical communication, cybersecurity, interoperability, and long-term maintenance performance.

Appendix A. Reproducible Training and Online Control Procedure

The following pseudo-code summarizes the offline training, online calibration, event-triggered edge-cloud coordination, command fusion, and safety projection using the notation of Sections 2.2–2.4.

Algorithm A1. Offline Training

1. Construct synchronized training windows. Interpolate valid samples using Eq. (10), impute missing values using Eq. (11), and compute input confidence using Eq. (12).
2. Build the topology-weighted graph using Eq. (13). Process the inputs through the GAT and temporal branches in Eqs. (14)–(18), and fuse their features using Eq. (19).
3. Predict the edge state, confidence, and risk using Eqs. (20)–(21), and compute the state-estimation, power-flow, and uncertainty-calibration losses.
4. For available cloud-training windows, generate multi-step forecasts using Eq. (23) and compute the prediction loss in Eq. (24).
5. Form the total objective in Eq. (37), update the trainable parameters using AdamW, and apply early stopping based on the validation score. After training, prune and quantize the edge model and freeze the backbone for online operation.

Algorithm A2. Online EC-HDT Control at Time t

6. Acquire multi-source measurements, synchronize timestamps, update masks and confidence values, and estimate the edge state x_t^e, confidence c_t, and risk r_t.

7. Evaluate the event trigger δ_t in Eq. (34). If triggered, upload the compressed edge state to the cloud, which updates the global state, generates H-step forecasts, and solves the MPC problem in Eqs. (25)–(29).

8. Compute the communication latency τ_t and age of information a_t. Normalize τ_t, a_t, c_t, r_t, and l_t, and evaluate the monotone cloud weight ω_t using Eq. (32).

9. Fuse the edge and cloud commands using Eq. (33), and project the fused command onto the feasible set U_t through the QP in Eq. (35). Execute only the first feasible action u_t^*.

10. Receive the next measurement and update only the bounded calibration variables according to Eq. (38). Accept the update if the recent residual decreases; otherwise retain θ_t.

11. $Set\ t \leftarrow t+1$ and return to Step 6.

References

[1] Aghazadeh Ardebili A, Zappatore M, Ramadan A I H A, et al. Digital Twins of smart energy systems: A systematic literature review on enablers, design, management and computational challenges[J]. Energy Informatics, 2024, 7: 94.

[2] Bayer D, Pruckner M. A digital twin of a local energy system based on real smart meter data[J]. Energy Informatics, 2023, 6: 8.

[3] Grieves M, Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems[C]//Kahlen F J, Flumerfelt S, Alves A, eds. Transdisciplinary Perspectives on Complex Systems. Cham: Springer, 2017: 85–113.

[4] Tao F, Zhang H, Liu A, Nee A Y C. Digital Twin in Industry: State-of-the-Art[J]. IEEE Transactions on Industrial Informatics, 2019, 15(4): 2405–2415.

[5] Das O, Zafar M H, Sanfilippo F, et al. Advancements in digital twin technology and machine learning for energy systems: A comprehensive review of applications in smart grids, renewable energy, and electric vehicle optimisation[J]. Energy Conversion and Management: X, 2024, 24: 100715.

[6] Sifat M M H, Das S K, Choudhury S M. Design, development, and optimization of a conceptual framework of digital twin electric grid using systems engineering approach[J]. Electric Power Systems Research, 2024, 226: 109958.

[7] Krishnan S B, Tan K T, Zhang W. Cognitive digital twin for microgrid: A real-world study for intelligent energy management and optimization[J]. IEEE Internet Computing, 2025, 29(1): 39-47.

[8] Dellaly M, Skander-Mustapha S, Slama-Belkhodja I. A digital twin model-based approach to cost optimization of residential community microgrids[J]. Global Energy Interconnection, 2024, 7(1): 82-93.

[9] [9] Thaler B, Posch S, Wimmer A, Pirker G. Hybrid model predictive control of renewable microgrids and seasonal hydrogen storage[J]. International Journal of Hydrogen Energy, 2023, 48(97): 38125–38142.

[10] Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge Computing: Vision and Challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637–646.

[11] Mao Y, You C, Zhang J, Huang K, Letaief K B. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322–2358.

[12] Li H, Yang Y, Liu Y, Pei W. Federated dueling DQN based microgrid energy management strategy in edge-cloud computing environment[J]. Sustainable Energy, Grids and Networks, 2024, 38: 101329.

[13] Habib M, Bollin E, Wang Q. Edge-based solution for battery energy management system: Investigating the integration capability into the building automation system[J]. Journal of Energy Storage, 2023, 72: 108479.

[14] Kundacina O, Cosovic M, Miskovic D, Vukobratovic D. Graph neural networks on factor graphs for robust, fast, and scalable linear state estimation with PMUs[J]. Sustainable Energy, Grids and Networks, 2023, 34: 101056.

[15] Ngo Q H, Nguyen B L H, Vu T V, et al. Physics-informed graphical neural network for power system state estimation[J]. Applied Energy, 2024, 358: 122602.

[16] Cibaku E, Gama F, Park S W. Boosting efficiency in state estimation of power systems by leveraging attention mechanism[J]. Energy and AI, 2024, 16: 100369.

[17] Yu J J Q, Wei X. ST-AGNet: Dynamic power system state prediction with spatial-temporal attention graph-based network[J]. Applied Energy, 2024, 365: 123252.

[18] Raghuvamsi Y, Teeparthi K, Kumar D M V, et al. Distribution system state estimation using physics-guided deep learning approach[J]. Electric Power Systems Research, 2024, 236: 110922.

[19] Casagrande V, Ferianc M, Rodrigues M R D, Boem F. Online end-to-end learning-based predictive control for microgrid energy management[J]. IEEE Transactions on Control Systems Technology, 2025, 33: 1-16.

[20] Nassereddine K, Turzynski M, Bielokha H, Strzelecki R. Simulation of energy management system using model predictive control in AC/DC microgrid[J]. Scientific Reports, 2025, 15: 5388.

[21] Lim S, Lee J, Lee S. Model predictive control-based energy management system for cooperative optimization of grid-connected microgrids[J]. Energies, 2025, 18(7): 1696.

[22] Kfouri M, et al. A resilient deep learning approach for state estimation in distribution grids with distributed generation[J]. International Transactions on Electrical Energy Systems, 2025: 2734170.

[23] Segura F, Andújar J M, et al. Methodology for energy management strategies design based on predictive control techniques for smart grids[J]. Applied Energy, 2023, 351: 121809.

[24] Sepehrzad R, Ghafourian J, Hedayatnia A, Al-Durra A, Khooban M H. Experimental and developed DC microgrid energy management integrated with battery energy storage based on multiple dynamic matrix model predictive control[J]. Journal of Energy Storage, 2023, 74: 109282.

[25] Kaul S, Yates R, Gruteser M. Real-Time Status: How Often Should One Update?[C]//IEEE INFOCOM. 2012: 2731–2735.

[26] Habibi S, Effatnejad R, Hedayati M, et al. Stochastic energy management of a microgrid incorporating two-point estimation method, mobile storage, and fuzzy multi-objective enhanced grey wolf optimizer[J]. Scientific Reports, 2024, 14: 1667.

[27] Mathur A, Kumari R, Meena V P, et al. Data-driven optimization for microgrid control under distributed energy resource variability[J]. Scientific Reports, 2024, 14: 10806.
[28] Keshta H E, Hassaballah E G, Ali A A, et al. Multi-level optimal energy management strategy for a grid tied microgrid considering uncertainty in weather conditions and load[J]. Scientific Reports, 2024, 14: 10059.
[29] Ahmed D, Ebeed M, Kamel S, et al. An enhanced jellyfish search optimizer for stochastic energy management of multi-microgrids with wind turbines, biomass and PV generation systems considering uncertainty[J]. Scientific Reports, 2024, 14: 15558.
[30] Elazab R, Abo Dahab A, Abo Adma M, et al. Reviewing the frontier: Modeling and energy management strategies for sustainable 100% renewable microgrids[J]. Discover Applied Sciences, 2024, 6: 168.
[31] Moshtagh S, Azimian B, Golgol M, Pal A. Topology-aware graph neural network-based state estimation for PMU-unobservable power systems[J]. IEEE Transactions on Power Systems, 2025, 40(6): 4547-4560.
[32] Chen Q, Bu S, Wang H, Lei C. Real-time multi-stability risk assessment and visualization of power systems: A graph neural network-based method[J]. IEEE Transactions on Power Systems, 2025, 40(4): 2955-2968.
[33] He G, Lei M, Han L, Zhao Y, Li Y. Energy-efficient cloud-edge collaborative model integrating digital twins and machine learning for scalable and adaptive distributed networks[J]. Sustainable Computing: Informatics and Systems, 2025, 47: 101157.
[34] Lee C, Kwon H, Lee Y I. OPC UA-based three-layer architecture for aggregated microgrids integrating edge cloud computing and IEC 62264[J]. Journal of Industrial Information Integration, 2025, 48: 100965.
[35] Vivas F J, Pajares A, Blasco X, Herrero J M, Segura F, Andújar J M. A novel energy management system based on two-level hierarchical economic model predictive control for use in microgrid control[J]. Energy Conversion and Management: X, 2025, 26: 101027.
[36] Su Z, Zheng G, Hu M, et al. Short-term load forecasting of regional integrated energy system based on spatio-temporal convolutional graph neural network[J]. Electric Power Systems Research, 2024, 232: 110427.
[37] Yang M, Qiu G, Liu T, et al. Probabilistic power flow based on physics-guided graph neural networks[J]. Electric Power Systems Research, 2024, 235: 110864.
[38] Wu P, Mei X. Microgrids energy management considering net-zero energy concept: The role of renewable energy landscaping design and IoT modeling in digital twin realistic simulator[J]. Sustainable Energy Technologies and Assessments, 2024, 63: 103621.
[39] Kumar M, Tyagi B. A machine learning-based stochastic optimal energy management framework for a renewable energy-assisted isolated microgrid system[J]. Energy Sources, Part B: Economics, Planning, and Policy, 2024, 19(1): 2294869.
[40] Tan M, Wang L, et al. Federated reinforcement learning for smart and privacy-preserving energy management of residential microgrids clusters[J]. Engineering Applications of Artificial Intelligence, 2025, 139: 109579.
[41] Khan N H, Wang Y, Jamal R, et al. A novel modified artificial rabbit optimization for stochastic energy management of a grid-connected microgrid: A case study in China[J]. Energy Reports, 2024, 11: 5436-5455.
[42] Jena C J, Ray P K. Power allocation scheme for grid interactive microgrid with hybrid energy storage system using model predictive control[J]. Journal of Energy Storage, 2024, 81: 110401.