EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

TC-LEG: Topology-Constrained Sequential Energy Management in Digital-Twin Smart Grids

Yuxiang Yang[1], Weijie Cheng[1], Zhi Li[1], Jiwei Gou[1,*], Yifan Chen[1]

[1]Power Dispatching and Control Center, Shenzhen Power Supply Bureau Co., Ltd., China Southern Power Grid, Shenzhen, Guangdong, 518001, China

Abstract

INTRODUCTION: The increasing integration of renewable energy, distributed storage, and flexible loads has introduced substantial uncertainty and operational complexity into modern smart grids. Existing energy management methods can optimize multi-period dispatch or encode the physical grid topology, but they often do not explicitly represent the execution order and conditional dependencies among heterogeneous control actions.

OBJECTIVES: This paper aims to develop a safe and energy-efficient sequential control generation framework for digital-twin smart grids that reduces operating cost and renewable curtailment while maintaining voltage security and action feasibility.

METHODS: A topology-constrained latent execution graph learning framework, termed TC-LEG, is proposed. TC-LEG uses a physical grid graph to encode electrical connectivity and operating states, while a distinct latent execution graph represents state-dependent dependencies among storage dispatch, renewable curtailment, demand response, reactive compensation, and topology switching. A hybrid discrete–continuous control sequence generator produces multi-step actions, and a simulation-oriented digital twin synchronizes the current grid state, verifies each candidate action, projects infeasible actions onto a state-dependent feasible set, and feeds the accepted action and updated state back to subsequent generation steps.

RESULTS: On the IEEE 33-bus system, TC-LEG achieves a normalized operating cost of 0.811, a voltage violation rate of 0.9%, a renewable curtailment rate of 4.9%, and an action feasibility rate of 98.5%. Its average inference time is 18.6 ms on the IEEE 33-bus system and 28.4 ms on the IEEE 69-bus system, both substantially shorter than the 15-minute control interval.

CONCLUSION: TC-LEG provides a topology-aware and interpretable approach to sequential energy management by combining electrical-topology representation, action-dependency learning, hybrid control generation, and digital-twin verification in a closed-loop inference process.

Received on 26 June 2026; accepted on 13 August 2026; published on 03 September 2026

Keywords: Smart grid; Digital twin; Energy management; Latent execution graph; Sequential control sequence generation; Hybrid energy storage; Renewable energy integration

doi:10.4108/ew.13744

1. Introduction

The increasing penetration of renewable energy resources, distributed storage systems, and flexible loads is reshaping the operation of modern power grids[1]. Compared with conventional power systems, renewable-integrated smart grids face stronger uncertainty, faster power fluctuations, and more complex coupling among generation, storage, load, and network topology[2]. Therefore, energy management systems (EMSs) are required not only to reduce operating cost, but also to maintain voltage security, avoid line overload, improve renewable energy utilization, and coordinate heterogeneous storage resources under strict physical constraints[3].

Digital twin technologies provide a promising foundation for smart energy systems by linking synchronized measurements, a continuously updated

*Corresponding author. Email: 151118031991@163.com

virtual grid model, state-transition simulation, and control verification in a bidirectional decision loop[4]. In this work, the digital twin is simulation-oriented: its operating state is synchronized at each control interval using the latest grid measurements, candidate actions are evaluated through topology checking and power-flow-based state transition, and the accepted action and resulting virtual state are returned to the generator. The current implementation uses offline-calibrated device and network parameters rather than continuous online parameter identification; this scope is stated explicitly to distinguish the proposed module from a generic static simulator.

In practical grid operation, a control decision is rarely an isolated action[5]. For example, when photovoltaic output suddenly drops, a supercapacitor may first provide fast compensation, a battery may then support sustained power balance, and demand response or topology reconfiguration may be activated if voltage or loading constraints remain violated. Such a process forms a multi-step control sequence[6]. Conventional multi-period dispatch optimizes control vectors over a horizon and correctly accounts for temporal state transitions, but it does not necessarily expose action-level execution order or the conditional dependency by which one action enables, constrains, or changes the feasible range of a subsequent action. TC-LEG therefore models both inter-temporal system evolution and directed dependencies among individual control-action tokens. This distinction is particularly relevant when switch operations alter network topology, battery dispatch changes the feasible reactive-power range, or fast storage actions must precede slower balancing actions.

Artificial intelligence methods have been widely studied for energy management and power grid control[7]. Reinforcement learning (RL) can improve online decision-making efficiency, while graph neural networks (GNNs) are well suited for modeling power grids because buses, loads, generators, storage units, and substations naturally form graph-structured systems[8]. Recent graph-based RL studies have shown promising potential in topology-aware power grid control[9]. Nevertheless, most existing methods mainly encode the physical grid topology and map the current state to a control vector[10]. They usually do not explicitly represent directed execution dependencies among individual actions, such as the temporal coordination between supercapacitors and batteries, the dependency between topology switching and line loading, and the relationship between active-power dispatch and reactive compensation[11].

Moreover, practical smart grid control involves hybrid discrete–continuous actions. Continuous actions include battery charging/discharging power, supercapacitor output, reactive power compensation, and renewable curtailment ratio, whereas discrete actions include switch operation, storage mode selection, and demand response activation[12]. Traditional optimization methods such as optimal power flow (OPF), mixed-integer programming (MIP), and model predictive control (MPC) can explicitly handle physical constraints, but their computational burden increases rapidly with grid scale, uncertainty, and action complexity[13]. Learning-based methods[14–16] provide faster online inference, but may generate physically infeasible or unsafe actions if topology constraints and execution dependencies are not properly modeled.

To address these challenges, this paper proposes TC-LEG, a topology-constrained latent execution graph learning framework for sequential energy management in digital-twin smart grids. The physical grid graph represents buses, branches, controllable devices, electrical connectivity, and operating states. In contrast, the latent execution graph is an action-level directed graph whose edges represent state-dependent conditional dependencies among control tokens. A hybrid discrete–continuous generator then produces a multi-step control sequence under topology-aware masks and device bounds. Finally, a digital-twin-in-the-loop verification module checks power-flow feasibility, voltage and line-loading constraints, storage limits, and topology feasibility; any corrected action and the resulting state are fed back to condition the next generation step.

The main contributions of this paper are summarized as follows:

- A physical-grid encoder is developed to represent bus–branch electrical connectivity, device locations, branch attributes, and the current operating state, thereby providing topology-dependent context for control generation.

- A distinct latent execution graph is introduced at the action-token level. Its directed weighted edges encode which preceding control actions condition subsequent actions, rather than duplicating the electrical connections represented by the physical grid graph.

- A hybrid discrete–continuous sequence generator combines execution-graph message passing, topology-aware discrete masks, and state-dependent continuous bounds to generate ordered actions for storage dispatch, renewable curtailment, demand response, reactive compensation, and topology switching.

- A simulation-oriented digital-twin verification loop evaluates every candidate action, applies a defined feasible-set projection when necessary, updates the virtual grid state, and feeds the accepted action back into subsequent decoding. Experiments on the IEEE 33-bus and

IEEE 69-bus systems validate the resulting cost, security, renewable-utilization, and inference-efficiency performance.

2. Related Work

2.1. Digital Twins for Smart Energy Systems

Digital twin technologies have received increasing attention in smart energy systems due to their ability to connect physical power grids with virtual models through synchronized data, state estimation, simulation, and control verification[17]. By integrating measurements from supervisory control and data acquisition (SCADA) systems, phasor measurement units (PMUs), smart meters, and Internet of Things (IoT) devices, digital twins can provide dynamic representations of grid operating conditions and support monitoring, fault diagnosis, predictive maintenance, and decision validation[4]. In renewable-integrated grids, digital twins are particularly useful for evaluating the impacts of intermittent generation, load uncertainty, and topology changes before applying control actions to the physical system[18]. However, most existing studies emphasize perception-oriented functions, such as monitoring, state estimation, and what-if analysis, or use the virtual model only for post-decision validation[19]. They provide limited support for action-by-action feedback during sequence generation. TC-LEG addresses this limitation by synchronizing the virtual state at each control interval, verifying each candidate action, and returning the accepted action and updated state to the generator. The present implementation uses offline-calibrated models and does not claim continuous online parameter identification.

2.2. AI-Based Energy Management and Hybrid Energy Storage Optimization

Artificial intelligence has been widely applied to energy management problems, including economic dispatch, demand response, renewable energy scheduling, and storage control[18, 20]. Compared with rule-based strategies and conventional optimization methods, learning-based approaches can improve online decision-making efficiency after training and adapt to complex operating conditions[21]. Hybrid energy storage systems further increase the flexibility of smart grids by combining storage devices with different response characteristics. For example, batteries are suitable for medium- and long-term energy balancing, while supercapacitors or flywheels can provide fast power compensation for short-term fluctuations[22]. Existing studies have investigated AI-powered storage dispatch to reduce operating cost, smooth renewable power, and improve grid stability[19, 23]. Nevertheless, many existing methods treat storage control as an isolated dispatch problem and do not sufficiently consider its coupling with grid topology, voltage regulation, renewable curtailment, and demand response[24]. The coordination between fast-response and high-capacity storage devices is also commonly embedded in fixed rules, which limits adaptation when the grid state or topology changes. TC-LEG instead represents these couplings as state-dependent action-level dependencies and learns how fast storage, sustained battery dispatch, curtailment, and network controls should condition one another.

2.3. Graph Learning for Power Grid Control

Power grids are naturally graph-structured systems, where buses, generators, loads, substations, and storage units can be represented as nodes, while transmission lines, transformers, and switches can be modeled as edges[25]. Graph neural networks and graph reinforcement learning have therefore been introduced for power system state estimation, voltage control, topology optimization, and emergency control[9]. By exploiting the physical bus-branch topology, graph-based models can capture spatial dependencies among grid components and improve the generalization ability of control policies under different operating conditions[26]. However, existing graph learning methods mainly focus on encoding the physical grid topology and therefore describe where electrical interactions occur, rather than how control actions depend on one another during execution. A direct state-to-action mapping may capture correlations implicitly, but it does not expose whether a switch operation, storage action, or reactive-compensation action should precede or constrain another control. TC-LEG separates these two roles: the physical graph represents electrical connectivity, whereas the latent execution graph learns directed, state-dependent dependencies among action tokens.

2.4. Sequential Decision-Making for Grid Control

Sequential decision-making methods, including model predictive control, reinforcement learning, and sequence generation models, have been studied for power grid control and energy management[27, 28]. Model predictive control can optimize future actions over a rolling horizon while considering system constraints, but it may become computationally expensive for large-scale grids with mixed discrete-continuous actions and uncertain renewable generation[29]. Reinforcement learning methods can learn control policies through interactions with simulated environments and provide fast online inference, but they may generate unsafe actions if physical constraints are not fully embedded during training or execution[30]. Recent sequence models, such as Transformer-based policies, provide a flexible way to generate multi-step action sequences[31]. However, directly applying generic sequence models to power grid control may produce infeasible actions because grid operation is constrained by topology, power flow, voltage limits, line capacities, and storage state-of-charge constraints. Moreover, conventional horizon-based optimization and generic sequence modeling do not necessarily expose the directed dependency among individual action tokens. TC-LEG combines explicit action-order modeling with topology-aware masking and digital-twin feedback so that the generated sequence is both economically coordinated and physically executable.

3. Methodology

3.1. Problem Definition and Overall Framework

We consider a renewable-integrated smart grid with distributed renewable generators, hybrid energy storage systems, flexible loads, and controllable switches. Throughout this paper, ordinary lowercase letters denote scalars, bold lowercase letters denote vectors, bold uppercase letters denote matrices, and calligraphic letters denote sets or graphs. A hat denotes a generated candidate, a tilde denotes an action accepted after feasibility verification or correction, and the superscript “exp” denotes an expert target used for training. Dimensions are stated when variables are first introduced.

The physical grid is modeled as

$$\mathcal{G}_{\mathrm{p}} = (\mathcal{V}, \mathcal{E}), \tag{1}$$

where $\mathcal{V}$ denotes the set of grid nodes, including buses, loads, renewable generators, batteries, supercapacitors, and substations, and $\mathcal{E}$ denotes the set of physical branches, including lines, transformers, and switches. The physical graph describes electrical connectivity and is distinct from the action-level execution graph introduced later.

At time step t, the node feature matrix is denoted by

$$\mathbf{X}_t = [\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \ldots, \mathbf{x}_{N,t}]^\top \in \mathbb{R}^{N\times F}, \tag{2}$$

where $N = |\mathcal{V}|$ and $\mathbf{x}_{i,t} \in \mathbb{R}^F$ contains the operating state of node i, such as voltage magnitude, active/reactive power injection, load demand, renewable generation, and state of charge (SOC) if the node is connected to a storage device. The edge feature of branch $(i, j) \in \mathcal{E}$ is denoted as $\mathbf{e}_{ij,t}$, which includes line impedance, line capacity, switch status, and branch power flow.

The objective is to generate an ordered multi-step control sequence over a prediction horizon T_p:

$$\widehat{\mathbf{A}}_{t:t+T_p-1} = [\widehat{\mathbf{a}}_t, \widehat{\mathbf{a}}_{t+1}, \ldots, \widehat{\mathbf{a}}_{t+T_p-1}]^\top, \tag{3}$$

where $\widehat{\mathbf{A}}_{t:t+T_p-1} \in \mathbb{R}^{T_p\times d_a}$ denotes the candidate sequence and $\widehat{\mathbf{a}}_\tau \in \mathbb{R}^{d_a}$ denotes one hybrid action vector. Each action consists of a discrete part and a continuous part:

$$\widehat{\mathbf{a}}_\tau = \left(\widehat{\mathbf{a}}^{\mathrm{d}}_\tau, \widehat{\mathbf{a}}^{\mathrm{c}}_\tau\right), \quad \tau = t, \ldots, t + T_p - 1. \tag{4}$$

Here, $\widehat{\mathbf{a}}^{\mathrm{d}}_\tau$ represents discrete decisions, such as switch operation, storage mode selection, and demand response activation, while $\widehat{\mathbf{a}}^{\mathrm{c}}_\tau$ represents continuous decisions, such as battery charging/discharging power, supercapacitor output, reactive power compensation, and renewable curtailment ratio.

TC-LEG is designed to generate topology-feasible and energy-efficient control sequences for digital-twin smart grids. As shown in the simplified high-level flow in Fig. 1, it contains four functionally distinct components: a topology-aware physical-grid encoder, a latent execution-graph learner, a hybrid discrete–continuous sequence generator, and a digital-twin verification and feedback module. The physical encoder represents electrical connections and operating states; the execution graph represents action dependencies; the generator produces ordered candidate actions; and the digital twin verifies or corrects each action before the accepted action and updated state condition the next decoding step.

3.2. Topology-Aware Grid Representation

The first component encodes the physical topology and real-time operating state of the power grid. Since the grid topology directly determines the feasible propagation of power flow and the validity of control actions, we employ an edge-aware graph encoder to obtain topology-aware node representations.

The initial node embedding is obtained by

$$\mathbf{h}^{(0)}_{i,t} = \mathbf{W}_x\mathbf{x}_{i,t} + \mathbf{b}_x, \tag{5}$$

where $\mathbf{W}_x$ and $\mathbf{b}_x$ are learnable parameters. For the l-th graph layer, the message from node j to node i is computed by considering both node and edge features:

$$\mathbf{m}^{(l)}_{ij,t} = \phi\left(\mathbf{W}^{(l)}_h\mathbf{h}^{(l)}_{j,t} + \mathbf{W}^{(l)}_e\mathbf{e}_{ij,t}\right), \tag{6}$$

where $\phi(\cdot)$ is a nonlinear activation function, and $\mathbf{W}^{(l)}_h$ and $\mathbf{W}^{(l)}_e$ are learnable transformation matrices. The node representation is then updated as

$$\mathbf{h}^{(l+1)}_{i,t} = \mathrm{LN}\left(\mathbf{h}^{(l)}_{i,t} + \sum_{j\in\mathcal{N}(i)} \alpha^{(l)}_{ij,t}\mathbf{m}^{(l)}_{ij,t}\right), \tag{7}$$

where $\mathcal{N}(i)$ is the neighbor set of node i, $\mathrm{LN}(\cdot)$ denotes layer normalization, and $\alpha^{(l)}_{ij,t}$ is the attention weight

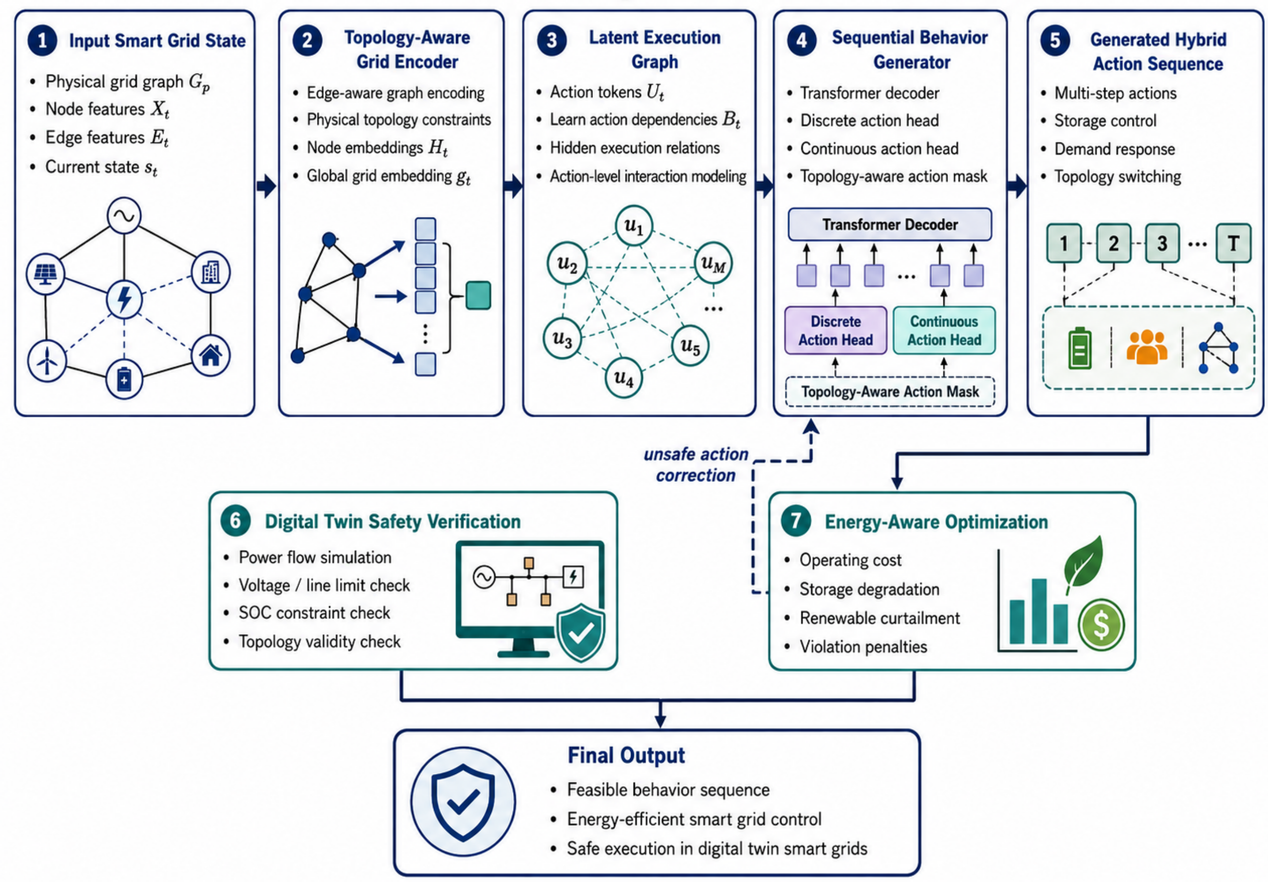

Figure 1. Functional flow of TC-LEG: physical-grid encoding, latent execution-graph learning, hybrid control sequence generation, and digital-twin verification with action and state feedback

between nodes i and j:

$$\alpha_{ij,t}^{(l)} = \frac{\exp\left(\psi\left(\mathbf{h}_{i,t}^{(l)}, \mathbf{h}_{j,t}^{(l)}, \mathbf{e}_{ij,t}\right)\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(\psi\left(\mathbf{h}_{i,t}^{(l)}, \mathbf{h}_{k,t}^{(l)}, \mathbf{e}_{ik,t}\right)\right)}, \tag{8}$$

where $\psi(\cdot)$ is a learnable attention scoring function.

After L_g graph layers, the topology-aware node representation is

$$\mathbf{H}_t = [\mathbf{h}_{1,t}, \mathbf{h}_{2,t}, \dots, \mathbf{h}_{N,t}]^\top \in \mathbb{R}^{N\times d}, \tag{9}$$

where $\mathbf{h}_{i,t} = \mathbf{h}_{i,t}^{(L_g)}$ and d is the hidden dimension. A global grid representation is further obtained by attention pooling:

$$\mathbf{g}_t = \sum_{i=1}^{N} \beta_{i,t}\mathbf{h}_{i,t}, \quad \beta_{i,t} = \frac{\exp(\mathbf{w}_p^\top \mathbf{h}_{i,t})}{\sum_{j=1}^{N} \exp(\mathbf{w}_p^\top \mathbf{h}_{j,t})}, \tag{10}$$

where $\mathbf{w}_p$ is a learnable pooling vector. The representations $\mathbf{H}_t$ and $\mathbf{g}_t$ provide the physical topology context for subsequent control sequence generation.

3.3. Latent Execution Graph-Guided Control Sequence Generation

Although the physical grid graph describes electrical connections, it does not explicitly represent execution dependencies among control actions. We therefore introduce a distinct latent execution graph

$$\mathcal{G}_{e,t} = (\mathcal{Q}, \mathcal{E}_{e,t}), \tag{11}$$

where $\mathcal{Q} = \{q^1, q^2, \dots, q^K\}$ is the set of K action tokens and $\mathcal{E}_{e,t}$ is the set of directed, state-dependent action dependencies. An edge $q^i \to q^j$ indicates that action token i is permitted to condition the generation or feasibility of action token j; it does not represent a physical transmission line or, by itself, establish a direct causal relationship.

Each action token is associated with a physical node and an action type. Let $\pi(k)$ denote the physical node related to token q^k, and let $\mathbf{r}_k$ denote its action-type embedding. The initial action token representation is defined as

$$\mathbf{u}_{k,t} = \mathbf{W}_u \left[\mathbf{h}_{\pi(k),t} \| \mathbf{g}_t \| \mathbf{r}_k\right], \tag{12}$$

where $\|$ denotes vector concatenation and $\mathbf{W}_u$ is a learnable projection matrix. The action token matrix is

$$\mathbf{U}_t = [\mathbf{u}_{1,t}, \mathbf{u}_{2,t}, \dots, \mathbf{u}_{K,t}]^\top \in \mathbb{R}^{K\times d}. \tag{13}$$

The latent execution dependency matrix is learned by masked self-attention:

$$\mathbf{B}_t = \mathrm{softmax}\left(\frac{(\mathbf{U}_t\mathbf{W}_Q)(\mathbf{U}_t\mathbf{W}_K)^\top}{\sqrt{d}} + \mathbf{M}_{\mathrm{e},t}\right), \tag{14}$$

where $\mathbf{B}_t \in [0,1]^{K\times K}$ is the learned dependency matrix, $\mathbf{W}_Q$ and $\mathbf{W}_K$ are learnable projection matrices, and $\mathbf{M}_{\mathrm{e},t} \in \{0, -\infty\}^{K\times K}$ is the execution feasibility mask. For a potential directed dependency $q^i \to q^j$, we first construct the binary admissibility indicator

$$c_{ij,t} = c_{ij,t}^{\mathrm{loc}} \vee c_{ij}^{\mathrm{dev}} \vee c_{ij,t}^{\mathrm{rule}}, \tag{15}$$

where $c_{ij,t}^{\mathrm{loc}} = 1$ when the two actions affect the same bus, an electrically adjacent bus, or a state-dependent influence region identified from the current physical topology; $c_{ij}^{\mathrm{dev}} = 1$ when the device-type compatibility table permits the action pair; and $c_{ij,t}^{\mathrm{rule}} = 1$ when an operational rule or expert constraint permits action i to precede or condition action j. The mask used by row j of the attention matrix is then

$$[\mathbf{M}_{\mathrm{e},t}]_{ji} = \begin{cases} 0, & c_{ij,t} = 1, \\ -\infty, & c_{ij,t} = 0. \end{cases} \tag{16}$$

Thus, $[\mathbf{B}_t]_{ji}$ is the normalized dependency weight from predecessor token q^i to target token q^j. The mask is reconstructed whenever switching changes the topology or device availability.

For example, battery active-power dispatch at a bus is allowed to condition reactive compensation at the same bus or its electrical neighborhood because it changes the local power balance and voltage profile. A switch operation is allowed to condition subsequent battery and reactive-compensation actions in the affected feeder because it changes connectivity and feasible power-flow paths. By contrast, actions assigned to electrically separated regions with no device compatibility or operational rule are masked from direct dependency. This construction combines automatically derived topology and device information with a small, auditable rule table rather than relying on an undefined notion of "influence."

The action representations are updated by message passing on the latent execution graph:

$$\widetilde{\mathbf{U}}_t = \mathrm{LN}\left(\mathbf{U}_t + \mathbf{B}_t\mathbf{U}_t\mathbf{W}_V\right), \tag{17}$$

where $\mathbf{W}_V$ is a learnable value projection matrix. A large value of $[\mathbf{B}_t]_{ji}$ means that the representation and generation of action j rely strongly on action i under the current grid state. Operators can inspect high-weight paths to identify dominant action ordering, diagnose an infeasible downstream action by tracing its upstream dependencies, and compare how control coordination changes across operating conditions. These weights are interpreted as learned conditional execution dependencies rather than direct physical causality.

Based on $\widetilde{\mathbf{U}}_t$, a Transformer decoder generates the future control sequence:

$$\mathbf{z}_\tau = \mathrm{Dec}_\theta\left(\mathbf{y}_{<\tau}, \widetilde{\mathbf{U}}_t, \mathbf{p}_\tau\right), \quad \tau = 1, \dots, T_p, \tag{18}$$

where $\mathbf{z}_\tau$ is the decoder hidden state, $\mathbf{y}_{<\tau}$ denotes previously generated action embeddings, and $\mathbf{p}_\tau$ is the temporal positional embedding.

For discrete actions, the output distribution is

$$\mathbf{p}_\tau^{\mathrm{d}} = \mathrm{softmax}\left(\mathbf{W}_d\mathbf{z}_\tau + \mathbf{b}_d + \mathbf{M}_\tau^{\mathrm{d}}\right), \tag{19}$$

where $\mathbf{M}_\tau^{\mathrm{d}}$ is a topology-aware discrete action mask that excludes infeasible actions, such as switching non-controllable branches or activating storage actions at non-storage nodes.

For continuous actions, the generator predicts bounded control variables:

$$\widehat{\mathbf{a}}_\tau^{\mathrm{c}} = \mathbf{a}_\tau^{\min} + \left(\mathbf{a}_\tau^{\max} - \mathbf{a}_\tau^{\min}\right) \odot \sigma\left(\mathbf{W}_c\mathbf{z}_\tau + \mathbf{b}_c\right), \tag{20}$$

where $\sigma(\cdot)$ is the sigmoid function, $\odot$ denotes element-wise multiplication, and $\mathbf{a}_\tau^{\min}$ and $\mathbf{a}_\tau^{\max}$ are the lower and upper bounds determined by device capacities and current operating states. This bounded formulation ensures that continuous actions remain within admissible physical ranges.

3.4. Digital Twin Verification and Energy-Aware Optimization

To ensure physical feasibility, every candidate action is evaluated by a simulation-oriented digital twin before it is accepted. At the beginning of control interval t, the virtual state $\widetilde{\mathbf{s}}_t$ is synchronized with the latest available grid state $\mathbf{s}_t$ obtained from the monitoring and state-estimation layer. Let $\mathcal{F}_{\mathrm{DT}}(\cdot)$ denote the calibrated virtual-grid transition model, which performs topology checking, device-state updates, and power-flow calculation. For a candidate action $\widehat{\mathbf{a}}_\tau$, a provisional next state is

$$\widehat{\mathbf{s}}_{\tau+1} = \mathcal{F}_{\mathrm{DT}}\left(\widetilde{\mathbf{s}}_\tau, \widehat{\mathbf{a}}_\tau\right), \quad \widetilde{\mathbf{s}}_t = \mathbf{s}_t. \tag{21}$$

The module qualifies as a digital-twin component in the present framework because it combines state synchronization, a calibrated virtual representation, bidirectional interaction with the decision model, and repeated state updating during control generation. Continuous online parameter identification is outside the present scope; network and device parameters are calibrated offline and can be refreshed between operating periods when new data are available.

The physical violation score is defined as

$$\Phi_\tau = \Phi_\tau^V + \Phi_\tau^S + \Phi_\tau^{\mathrm{SOC}}, \tag{22}$$

$$\Phi_\tau^V = \sum_{i\in\mathcal{V}} \left[\left(V_i^{\min} - \widehat{V}_{i,\tau} \right)_+^2 + \left(\widehat{V}_{i,\tau} - V_i^{\max} \right)_+^2 \right], \tag{23}$$

$$\Phi_\tau^S = \sum_{(i,j)\in\mathcal{E}} \left(|\widehat{S}_{ij,\tau}| - S_{ij}^{\max} \right)_+^2, \tag{24}$$

$$\Phi_\tau^{\mathrm{SOC}} = \sum_{b\in\mathcal{B}} \left[\left(\mathrm{SOC}_b^{\min} - \widehat{\mathrm{SOC}}_{b,\tau} \right)_+^2 + \left(\widehat{\mathrm{SOC}}_{b,\tau} - \mathrm{SOC}_b^{\max} \right)_+^2 \right]. \tag{25}$$

where $\mathcal{B}$ is the set of battery units, $V_i^{\min}$ and $V_i^{\max}$ are voltage limits, $S_{ij}^{\max}$ is the branch capacity, and $\mathrm{SOC}_b^{\min}$ and $\mathrm{SOC}_b^{\max}$ are battery SOC limits. A control sequence is considered feasible if $\Phi_\tau = 0$ for all prediction steps and the resulting topology is connected, radial when required, and operationally valid.

Let $\Omega_\tau(\widetilde{\mathbf{s}}_\tau)$ denote the state-dependent feasible set containing the admissible discrete actions, device bounds, topology requirements, and power-flow constraints at step τ. The correction operator is explicitly defined as the nearest-feasible projection

$$\widetilde{\mathbf{a}}_\tau = \Pi_{\Omega_\tau}(\widehat{\mathbf{a}}_\tau) = \arg\min_{\mathbf{a}\in\Omega_\tau(\widetilde{\mathbf{s}}_\tau)} \|\mathbf{W}_\Omega(\mathbf{a} - \widehat{\mathbf{a}}_\tau)\|_2^2, \tag{26}$$

where $\mathbf{W}_\Omega$ balances changes to different action components. For the discrete component, infeasible choices are removed and the feasible candidate with the highest decoder probability is selected; a no-operation token is used if no nontrivial discrete action is feasible. For the continuous component, device-level variables are first clipped to their state-dependent bounds and the remaining coupled variables are adjusted by the least-distance constrained projection above. After correction, the digital twin is evaluated again using $\widetilde{\mathbf{a}}_\tau$, and the resulting accepted state $\widetilde{\mathbf{s}}_{\tau+1}$ replaces the provisional state. The corrected action embedding is appended to the decoder history, so all subsequent actions are conditioned on what would actually be executed rather than on the rejected candidate.

The model is trained using expert trajectories generated from historical operation records, conventional optimization solvers, or digital twin simulations. The total training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{d}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{c}} + \lambda_{\mathrm{e}}\mathcal{L}_{\mathrm{energy}} + \lambda_{\mathrm{g}}\mathcal{L}_{\mathrm{graph}}, \tag{27}$$

where $\mathcal{L}_{\mathrm{d}}$ is the cross-entropy loss for discrete actions, $\mathcal{L}_{\mathrm{c}}$ is the regression loss for continuous actions, $\mathcal{L}_{\mathrm{energy}}$

Algorithm 1 Closed-Loop Inference Procedure of TC-LEG

Require: Physical grid graph $\mathcal{G}_{\mathrm{p}}$, synchronized state $\mathbf{s}_t$, node features $\mathbf{X}_t$, edge features $\mathbf{E}_t$, horizon T_p

Ensure: Feasible accepted sequence $\widetilde{\mathbf{A}}_{t:t+T_p-1}$

Encode the physical grid: $\mathbf{H}_t, \mathbf{g}_t \leftarrow f_\theta(\mathcal{G}_{\mathrm{p}}, \mathbf{X}_t, \mathbf{E}_t)$
Construct action tokens $\mathbf{U}_t$ and execution mask $\mathbf{M}_{\mathrm{e},t}$.
Learn $\mathbf{B}_t \leftarrow \mathrm{LEG}(\mathbf{U}_t, \mathbf{M}_{\mathrm{e},t})$ and update $\widetilde{\mathbf{U}}_t \leftarrow \mathrm{GraphUpdate}(\mathbf{U}_t, \mathbf{B}_t)$.
Initialize accepted virtual state $\widetilde{\mathbf{s}}_t \leftarrow \mathbf{s}_t$ and decoder history $\mathbf{y}_{<1} \leftarrow \varnothing$.
for $\tau = 1$ to T_p **do**
    Update state-dependent discrete mask, continuous bounds, and feasible set $\Omega_{t+\tau-1}(\widetilde{\mathbf{s}}_{t+\tau-1})$.
    Generate decoder state: $\mathbf{z}_\tau \leftarrow \mathrm{Dec}_\theta(\mathbf{y}_{<\tau}, \widetilde{\mathbf{U}}_t, \mathbf{p}_\tau)$.
    Generate candidate discrete and continuous components: $\widehat{\mathbf{a}}^{\mathrm{d}}_{t+\tau-1} \leftarrow \mathrm{MaskedDecode}(\mathbf{z}_\tau, \mathbf{M}^{\mathrm{d}}_{t+\tau-1})$, $\widehat{\mathbf{a}}^{\mathrm{c}}_{t+\tau-1} \leftarrow \mathrm{BoundedRegressor}(\mathbf{z}_\tau)$.
    Form candidate hybrid action $\widehat{\mathbf{a}}_{t+\tau-1}$ and simulate provisional state $\widehat{\mathbf{s}}_{t+\tau} \leftarrow \mathcal{F}_{\mathrm{DT}}(\widetilde{\mathbf{s}}_{t+\tau-1}, \widehat{\mathbf{a}}_{t+\tau-1})$.
    if $\widehat{\mathbf{a}}_{t+\tau-1} \notin \Omega_{t+\tau-1}$ or the provisional state violates physical/topology constraints **then**
        Project candidate onto the feasible set: $\widetilde{\mathbf{a}}_{t+\tau-1} \leftarrow \Pi_{\Omega_{t+\tau-1}}(\widehat{\mathbf{a}}_{t+\tau-1})$.
    else
        $\widetilde{\mathbf{a}}_{t+\tau-1} \leftarrow \widehat{\mathbf{a}}_{t+\tau-1}$.
    end if
    Re-evaluate and accept the next virtual state: $\widetilde{\mathbf{s}}_{t+\tau} \leftarrow \mathcal{F}_{\mathrm{DT}}(\widetilde{\mathbf{s}}_{t+\tau-1}, \widetilde{\mathbf{a}}_{t+\tau-1})$.
    Append accepted action to decoder history: $\mathbf{y}_\tau \leftarrow \mathrm{Embed}(\widetilde{\mathbf{a}}_{t+\tau-1})$.
end for
return $\widetilde{\mathbf{A}}_{t:t+T_p-1}$

---

is the energy-aware operating cost loss, and $\mathcal{L}_{\mathrm{graph}}$ is a sparsity regularization term for the latent execution graph.

The discrete action loss is

$$\mathcal{L}_{\mathrm{d}} = -\frac{1}{T_p} \sum_{\tau=1}^{T_p} \sum_{k=1}^{K_d} y^{\mathrm{d}}_{\tau,k} \log p^{\mathrm{d}}_{\tau,k}, \tag{28}$$

where K_d is the number of discrete action categories, $y^{\mathrm{d}}_{\tau,k}$ is the expert label, and $p^{\mathrm{d}}_{\tau,k}$ is the predicted probability. The continuous action loss is

$$\mathcal{L}_{\mathrm{c}} = \frac{1}{T_p} \sum_{\tau=1}^{T_p} \left\| \widehat{\mathbf{a}}^{\mathrm{c}}_\tau - \mathbf{a}^{\mathrm{c,exp}}_\tau \right\|_2^2, \tag{29}$$

where $\mathbf{a}^{\mathrm{c,exp}}_\tau$ denotes the expert continuous action.

The energy-aware loss is defined as

$$\mathcal{L}_{\text{energy}} = \frac{1}{T_p} \sum_{\tau=1}^{T_p} \left(C_\tau^{\text{grid}} + C_\tau^{\text{deg}} + C_\tau^{\text{curt}} + C_\tau^{\text{vio}} + C_\tau^{\text{sw}} \right), \tag{30}$$

where C_τ^{grid} is the grid purchasing cost, C_τ^{deg} is the storage degradation cost, C_τ^{curt} is the renewable curtailment cost, C_τ^{vio} is the physical constraint violation cost, and C_τ^{sw} is the switch operation cost. The latent graph regularization term is

$$\mathcal{L}_{\text{graph}} = \frac{1}{K^2} \|\mathbf{B}_t\|_1, \tag{31}$$

which encourages sparse and interpretable execution dependencies.

During inference, TC-LEG encodes the synchronized physical-grid state, learns the state-dependent execution graph, and generates one candidate hybrid action at a time. The digital twin then verifies or corrects the candidate, advances the accepted virtual state, and returns both the accepted action and updated state to the decoder. This closed-loop design prevents later actions from being conditioned on a rejected candidate and supports energy-efficient, topology-feasible, and physically safe control sequences.

4. Experiments

4.1. Experimental Setup

To evaluate the effectiveness of the proposed TC-LEG framework, experiments are conducted on benchmark distribution systems with renewable generation, hybrid energy storage, flexible loads, and controllable switches. The experiments are designed to answer the following questions: 1) whether the proposed method can reduce the operating cost of smart grid energy management; 2) whether the generated control sequences can satisfy grid security constraints, especially voltage limits and line loading constraints; 3) whether the proposed method can improve renewable energy utilization and reduce renewable curtailment; 4) whether the proposed method can maintain robust performance under topology disturbances; and 5) how each key component contributes to energy-aware sequential control generation.

The proposed method is compared with representative rule-based, optimization-based, reinforcement learning-based, graph learning-based, and sequence generation-based methods. All methods are evaluated under the same load profiles, renewable generation profiles, storage configurations, electricity price settings, and topology disturbance scenarios. Unless otherwise specified, the renewable penetration level is set to 60%, the control interval is 15 minutes, and each test episode covers a 24-hour operation horizon with 96 time steps. The prediction horizon of the control sequence generator is set to $T_p = 12$.

4.2. Benchmark Systems and Scenario Construction

Benchmark power grid systems. Two standard distribution systems are used as the main test systems: the IEEE 33-bus system and the IEEE 69-bus system. The IEEE 33-bus system is used for the main comparison, sensitivity analysis, and ablation study, while the IEEE 69-bus system is used to evaluate scalability and topology generalization. Each system is augmented with distributed photovoltaic generation, batteries, supercapacitors, flexible loads, and controllable switches. The physical grid is modeled as a graph, where buses are treated as nodes and distribution lines or switches are treated as edges. Batteries are mainly used for medium- and long-term energy balancing, while supercapacitors are used for fast power fluctuation compensation.

The detailed system configuration is summarized in Table 1. The base load values follow the standard benchmark system settings, and additional renewable and storage resources are installed at selected buses to construct renewable-integrated smart grid scenarios.

Renewable generation and load scenarios. To evaluate the performance under different operating conditions, multiple scenarios are constructed by varying renewable penetration, load demand, storage capacity, and topology disturbance. The renewable penetration level is defined as the ratio between the installed renewable generation capacity and the peak load:

$$\gamma_{\text{ren}} = \frac{P_{\text{cap}}^{\text{ren}}}{P_{\text{peak}}^{\text{load}}}. \tag{32}$$

Four renewable penetration levels are considered:

$$\gamma_{\text{ren}} \in \{20\%, 40\%, 60\%, 80\%\}. \tag{33}$$

The load profiles include normal load, peak load, and stochastic load fluctuation scenarios. The renewable generation profiles include normal photovoltaic generation, fast photovoltaic drop caused by cloud cover, and stochastic renewable fluctuation. Gaussian noise is added to both load and renewable profiles to simulate realistic uncertainty. The standard deviation of load uncertainty is set to 5% of the nominal demand, and the standard deviation of renewable uncertainty is set to 8% of the available renewable generation. These settings are used to simulate renewable-integrated smart grid operation under uncertain supply-demand conditions.

Topology disturbance scenarios. To evaluate the robustness of different methods to topology changes, three topology disturbance settings are considered:

- **Normal topology**: the original benchmark topology is used.

Table 1. Configuration of benchmark power grid systems

Configuration	IEEE 33-bus	IEEE 69-bus
Number of buses	33	69
Number of branches	32 + 5 tie lines	68 + 5 tie lines
Base load	3.72 MW / 2.30 MVAr	3.80 MW / 2.69 MVAr
Renewable generators	4 PV units	7 PV units
PV installation buses	6, 14, 25, 30	11, 18, 26, 41, 50, 61, 66
Battery units	2	3
Battery buses	14, 30	21, 50, 61
Supercapacitor units	2	3
Supercapacitor buses	18, 25	18, 41, 66
Flexible load nodes	6	10
Controllable switches	8	10
Voltage limits	0.95–1.05 p.u.	0.95–1.05 p.u.
Battery SOC limits	0.20–0.90	0.20–0.90
Time interval	15 min	15 min
Simulation horizon	24 h / 96 steps	24 h / 96 steps
Prediction horizon T_p	12 steps	12 steps
Training / validation / test days	180 / 30 / 60	180 / 30 / 60

- **Single-line outage**: one branch is disconnected during operation while the system remains connected through feasible switching.
- **Switch reconfiguration**: controllable switches are changed to modify the power flow distribution while maintaining radiality and connectivity.

These scenarios are used to verify whether the proposed topology-constrained design can improve the feasibility and security of generated control sequences under changing grid structures. Such topology changes are common in distribution systems due to fault isolation, maintenance, congestion management, and service restoration.

4.3. Compared Methods

The proposed TC-LEG is compared with the following methods:

- **Rule-based EMS**: A rule-based energy management strategy based on predefined thresholds of renewable generation, load demand, and storage state of charge.
- **OPF**: An optimal power flow-based dispatch method that optimizes the current operating state under physical constraints.
- **MPC**: A model predictive control method that optimizes future actions over a rolling horizon.
- **PPO**: Proximal policy optimization, an RL method that learns a control policy through interactions with the simulated grid environment.
- **GNN-RL**: A GNN-based RL method that encodes the physical grid topology before action prediction.
- **Transformer Policy**: A sequence generation method that generates multi-step actions using a Transformer decoder without explicit topology constraints.
- **Topology-Masked Transformer (TMT)**: A Transformer-based control sequence generation model with topology-aware action masks but without the latent execution graph.
- **TC-LEG**: The proposed method, which jointly uses the physical grid graph, latent execution graph, topology-aware control sequence generation, and digital twin verification.

For fair comparison, all learning-based methods are trained using the same training scenarios and evaluated on the same test scenarios. The control horizon, simulation interval, storage configuration, and electricity price settings are kept identical for all methods. For learning-based methods, the reported results are averaged over five random seeds.

4.4. Evaluation Metrics

Three primary metrics are used to evaluate the economic efficiency, grid security, and renewable energy utilization of different methods.

Total operating cost. The total operating cost is defined as

$$C_{\text{total}} = \sum_{t=1}^{T} \left(C_t^{\text{grid}} + C_t^{\text{deg}} + C_t^{\text{curt}} + C_t^{\text{vio}} + C_t^{\text{sw}} \right), \quad (34)$$

where C_t^{grid} is the grid purchasing cost, C_t^{deg} is the storage degradation cost, C_t^{curt} is the renewable curtailment cost, C_t^{vio} is the physical constraint violation cost, and C_t^{sw} is the switch operation cost. To make the results comparable across different systems, C_{total} is normalized by the cost of Rule-based EMS in each benchmark system.

Voltage violation rate. The voltage violation rate is used to measure grid operational security:

$$R_{\mathrm{volt}} = \frac{\sum_{t=1}^{T} \sum_{i \in \mathcal{V}} \mathbb{I}\left(V_{i,t} < V_i^{\min} \text{ or } V_{i,t} > V_i^{\max}\right)}{T|\mathcal{V}|}, \quad (35)$$

where $\mathbb{I}(\cdot)$ is the indicator function. A lower voltage violation rate indicates that the generated control sequence better satisfies voltage security constraints.

Renewable curtailment rate. The renewable curtailment rate is defined as

$$R_{\mathrm{curt}} = \frac{\sum_{t=1}^{T} \sum_{r \in \mathcal{R}} P_{r,t}^{\mathrm{curt}}}{\sum_{t=1}^{T} \sum_{r \in \mathcal{R}} P_{r,t}^{\mathrm{ren}}}, \quad (36)$$

where $\mathcal{R}$ denotes the set of renewable generators, $P_{r,t}^{\mathrm{curt}}$ is the curtailed renewable power, and $P_{r,t}^{\mathrm{ren}}$ is the available renewable power. A lower renewable curtailment rate indicates better renewable energy utilization.

In addition, we report the action feasibility rate and inference time as auxiliary metrics. The action feasibility rate is defined as

$$R_{\mathrm{feas}} = \frac{N_{\mathrm{feasible}}}{N_{\mathrm{generated}}}, \quad (37)$$

where N_{feasible} is the number of feasible generated actions and $N_{\mathrm{generated}}$ is the total number of generated actions.

4.5. Main Comparison Results

Table 2 reports the main comparison results on the IEEE 33-bus and IEEE 69-bus systems. Overall, TC-LEG achieves the best energy management performance on both systems in terms of operating cost, voltage security, renewable energy utilization, and action feasibility. On the IEEE 33-bus system, TC-LEG reduces the normalized total operating cost to 0.811, outperforming the strongest sequence-generation baseline, TMT, whose cost is 0.843. This indicates that learning latent execution dependencies among heterogeneous control actions can improve the economic dispatch of renewable generation, hybrid storage, and flexible loads. Meanwhile, TC-LEG also obtains the lowest voltage violation rate of 0.9% and the lowest renewable curtailment rate of 4.9%, showing that the proposed method does not reduce cost by sacrificing grid security or renewable utilization. The overall cost-security-curtailment trade-off on the IEEE 33-bus system is further visualized in Fig. 2.

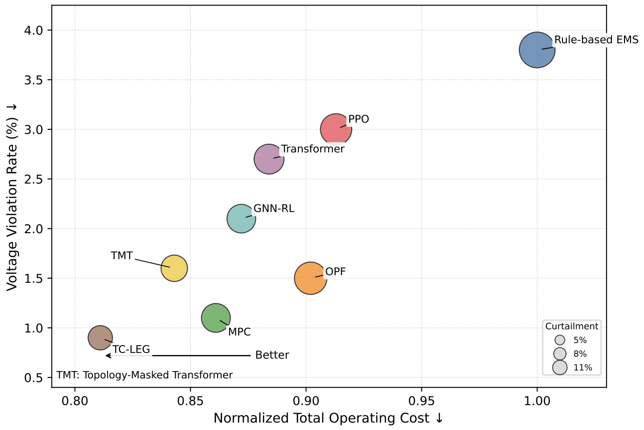

Figure 2. Cost-security-curtailment trade-off on the IEEE 33-bus system. The horizontal axis denotes the normalized total operating cost, the vertical axis denotes the voltage violation rate, and the bubble size represents the renewable curtailment rate. A better method is expected to be located closer to the lower-left region with a smaller bubble. TC-LEG achieves the best overall trade-off among economic efficiency, grid security, and renewable energy utilization

A similar trend can be observed on the IEEE 69-bus system. Although all methods suffer from performance degradation due to the larger network scale and more complex power flow interactions, TC-LEG still achieves the lowest normalized operating cost, voltage violation rate, and renewable curtailment rate. This result suggests that the proposed topology-aware graph representation and latent execution graph are effective not only on small-scale distribution systems but also on larger and more complex grid topologies. From an energy system perspective, these results demonstrate that TC-LEG can simultaneously improve cost efficiency, operational security, and renewable energy accommodation, which are three key objectives in renewable-integrated smart grid management.

As shown in Fig. 2, TC-LEG is located closest to the lower-left region and has a relatively small bubble size, indicating that it achieves a better balance among operating cost reduction, voltage security, and renewable curtailment mitigation. This visualization further confirms that the proposed method improves energy efficiency without compromising grid security or renewable energy accommodation.

Compared with Transformer Policy, TC-LEG significantly improves the action feasibility rate. This is mainly because generic sequence models lack explicit knowledge of physical grid topology and may generate

Table 2. Main comparison results on benchmark power grid systems. C_{total} is normalized by the cost of Rule-based EMS in each system. R_{volt}, R_{curt}, and R_{feas} are reported in percentage

Method	**IEEE 33-bus**				**IEEE 69-bus**			
	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{curt}} \downarrow$	$R_{\text{feas}} \uparrow$	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{curt}} \downarrow$	$R_{\text{feas}} \uparrow$
Rule-based EMS	1.000	3.8	10.6	92.1	1.000	5.1	13.2	89.3
OPF	0.902	1.5	8.7	99.5	0.919	2.3	10.4	99.3
MPC	0.861	1.1	6.9	99.1	0.881	2.0	8.9	98.5
PPO	0.913	3.0	8.1	93.5	0.944	4.4	11.6	89.9
GNN-RL	0.872	2.1	6.8	95.9	0.897	3.1	9.1	94.1
Transformer Policy	0.884	2.7	7.4	91.6	0.916	4.0	10.3	88.7
TMT	0.843	1.6	5.8	97.3	0.872	2.5	7.9	95.6
TC-LEG	**0.811**	**0.9**	**4.9**	**98.5**	**0.842**	**1.6**	**6.8**	**97.2**

invalid switching, storage, or curtailment actions. By introducing topology-aware action masks and physical grid graph representations, TC-LEG reduces infeasible actions and improves the executability of generated control sequences. Compared with TMT, TC-LEG further reduces operating cost and renewable curtailment, indicating that topology masking alone is insufficient for coordinated energy management. The latent execution graph helps model the dependencies among supercapacitor response, battery dispatch, renewable curtailment, voltage regulation, and topology switching, thereby improving multi-step energy coordination.

It is worth noting that OPF and MPC obtain high action feasibility because they explicitly solve constrained optimization problems. However, their operating costs are higher than that of TC-LEG in dynamic multi-step scenarios. This suggests that purely optimization-based methods may be conservative when handling uncertainty and hybrid discrete–continuous decisions. In contrast, TC-LEG learns from historical and simulated dispatch trajectories while preserving physical feasibility through digital twin verification, achieving a better balance between energy economy and grid security.

4.6. Inference Efficiency

Table 3 compares the average inference time of different methods, and Fig. 3 further illustrates the trade-off between inference efficiency and operating cost. Rule-based EMS has the shortest inference time, but its energy management performance is limited because it relies on fixed thresholds and cannot adaptively coordinate renewable generation, storage, and network constraints. OPF and MPC provide physically feasible dispatch solutions, but their computation time increases significantly with system scale. In particular, MPC requires 624.8 ms on the IEEE 33-bus system and 1518.5 ms on the IEEE 69-bus system, which may limit its applicability in fast online energy management when frequent re-dispatch is required.

Table 3. Average inference time of different methods

Method	**IEEE 33-bus**	**IEEE 69-bus**
Rule-based EMS	0.4 ms	0.5 ms
OPF	210.6 ms	587.3 ms
MPC	624.8 ms	1518.5 ms
PPO	4.8 ms	7.1 ms
GNN-RL	8.7 ms	13.4 ms
Transformer Policy	12.5 ms	17.9 ms
TMT	13.1 ms	18.5 ms
TC-LEG	18.6 ms	28.4 ms

Learning-based methods achieve much faster online inference after training. TC-LEG requires slightly more inference time than PPO, GNN-RL, Transformer Policy, and TMT because it additionally performs latent execution graph learning and digital twin-based safety verification. However, its inference time remains only 18.6 ms on the IEEE 33-bus system and 28.4 ms on the IEEE 69-bus system, which is far below the 15-minute dispatch interval used in the experiments. Therefore, the additional computation introduced by TC-LEG is acceptable for practical smart grid operation. More importantly, Fig. 3 shows that TC-LEG achieves the lowest operating cost while maintaining fast online inference, demonstrating a favorable trade-off between computational efficiency and energy management performance.

4.7. Robustness under Topology Disturbances

To evaluate robustness to topology changes, different methods are tested under normal topology, single-line outage, and switch reconfiguration scenarios on the IEEE 33-bus system. As shown in Table 4, topology disturbances generally increase operating cost and voltage violation while reducing action feasibility for all methods. This is because line outages and switch reconfiguration change the power flow distribution, making the coordination among storage dispatch, voltage regulation, and topology control more difficult.

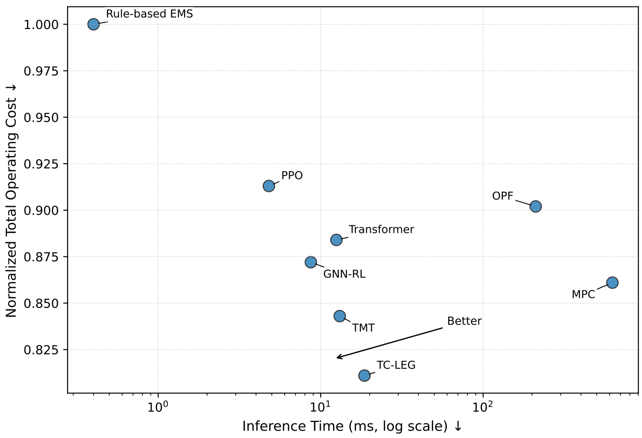

Figure 3. Inference time versus normalized total operating cost on the IEEE 33-bus system. The horizontal axis is shown in logarithmic scale to better distinguish methods with different computational costs. Although TC-LEG introduces additional computation for latent execution graph learning and digital twin verification, it remains much faster than OPF and MPC while achieving the lowest operating cost, demonstrating a favorable balance between online efficiency and energy management performance

Compared with the baseline methods, TC-LEG shows the smallest performance degradation under topology disturbances. Under the single-line outage scenario, the voltage violation rate of GNN-RL increases from 2.1% to 4.7%, while its action feasibility rate decreases from 95.9% to 89.8%. By contrast, TC-LEG maintains a voltage violation rate of 2.0% and an action feasibility rate of 96.3%. This indicates that TC-LEG can generate more reliable and executable control sequences even when the physical grid structure changes. The advantage mainly comes from two aspects: the physical grid graph encoder captures topology-dependent power flow patterns, while the topology-aware action mask and digital twin verification prevent unsafe switching and storage actions.

From the perspective of energy system operation, this robustness is important because renewable-integrated distribution systems may frequently experience topology changes caused by fault isolation, network reconfiguration, maintenance, or congestion management. A dispatch method that performs well only under a fixed topology may not be sufficient for practical smart grids. The results show that TC-LEG can maintain cost-effective and secure operation under topology disturbances, supporting its potential for digital-twin-based adaptive energy management.

4.8. Sensitivity Analysis

Sensitivity to renewable penetration. Fig. 4 and Table 5 report the sensitivity analysis under different renewable

Table 4. Robustness comparison under topology disturbances on the IEEE 33-bus system

Method	Scenario	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{feas}} \uparrow$
MPC	Normal	0.861	1.1	99.1
MPC	Single-line outage	0.913	2.8	95.8
MPC	Reconfiguration	0.887	1.9	96.9
GNN-RL	Normal	0.872	2.1	95.9
GNN-RL	Single-line outage	0.955	4.7	89.8
GNN-RL	Reconfiguration	0.921	3.5	92.6
TMT	Normal	0.843	1.6	97.3
TMT	Single-line outage	0.904	3.2	93.5
TMT	Reconfiguration	0.872	2.4	95.1
TC-LEG	Normal	**0.811**	**0.9**	**98.5**
TC-LEG	Single-line outage	**0.858**	**2.0**	**96.3**
TC-LEG	Reconfiguration	**0.833**	**1.4**	**97.4**

penetration levels. As the renewable penetration level increases from 20% to 60%, the total operating cost generally decreases because more low-cost renewable energy is available to serve the load. However, when the renewable penetration reaches 80%, the operating cost increases again for all methods. This is because excessive renewable generation introduces stronger power fluctuations and higher curtailment risk, and the additional voltage violation and curtailment penalties offset part of the economic benefit of renewable generation.

TC-LEG consistently achieves the lowest operating cost, voltage violation rate, and renewable curtailment rate across all renewable penetration levels. At 80% renewable penetration, TC-LEG reduces the curtailment rate to 10.5%, compared with 13.7% for MPC and 15.2% for GNN-RL. This demonstrates that TC-LEG can better coordinate battery storage, supercapacitor response, renewable curtailment, and topology-aware control actions under high-renewable operating conditions. The result is particularly relevant to future smart grids, where increasing renewable penetration requires energy management systems to absorb more renewable power while maintaining voltage security.

The trend also shows that the advantage of TC-LEG becomes more evident as renewable penetration increases. Under low-renewable scenarios, the dispatch problem is relatively simple, and the performance gap among methods is smaller. Under high-renewable scenarios, however, the system requires more complex coordination among fast-response storage, long-duration battery dispatch, and network constraints. The latent execution graph helps capture such action dependencies, enabling TC-LEG to reduce renewable curtailment and improve energy utilization more effectively.

Sensitivity to storage capacity. Table 6 presents the results under different storage capacity settings. Increasing storage capacity generally reduces operating

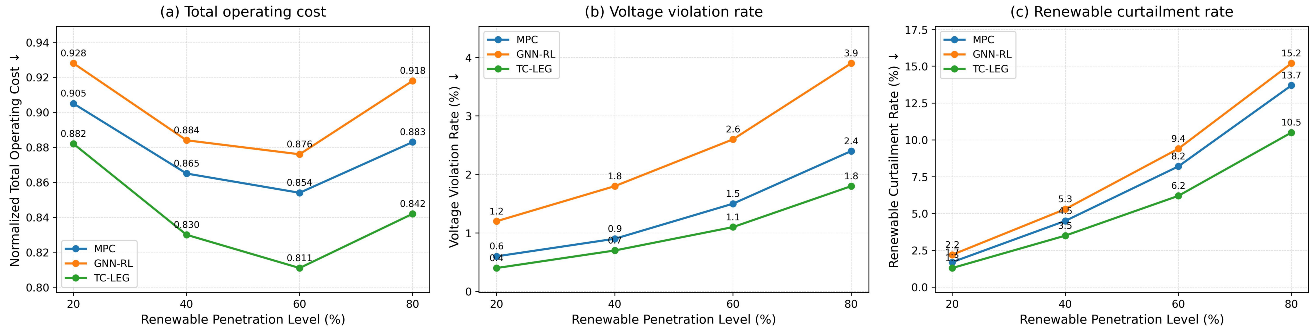

Figure 4. Sensitivity analysis under different renewable penetration levels on the IEEE 33-bus system. Three metrics are reported: normalized total operating cost, voltage violation rate, and renewable curtailment rate. As the renewable penetration level increases, all methods face stronger uncertainty and higher curtailment risk. TC-LEG consistently achieves lower operating cost, fewer voltage violations, and lower renewable curtailment than MPC and GNN-RL, demonstrating stronger adaptability to high-renewable smart grid operation

Table 5. Sensitivity analysis under different renewable penetration levels on the IEEE 33-bus system

Penetration	**Method**	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{curt}} \downarrow$
20%	MPC	0.905	0.6	1.7
20%	GNN-RL	0.928	1.2	2.2
20%	TC-LEG	**0.882**	**0.4**	**1.3**
40%	MPC	0.865	0.9	4.5
40%	GNN-RL	0.884	1.8	5.3
40%	TC-LEG	**0.830**	**0.7**	**3.5**
60%	MPC	0.854	1.5	8.2
60%	GNN-RL	0.876	2.6	9.4
60%	TC-LEG	**0.811**	**1.1**	**6.2**
80%	MPC	0.883	2.4	13.7
80%	GNN-RL	0.918	3.9	15.2
80%	TC-LEG	**0.842**	**1.8**	**10.5**

Table 6. Sensitivity analysis under different storage capacity settings on the IEEE 33-bus system

Capacity	**Method**	$C_{\text{total}} \downarrow$	$R_{\text{curt}} \downarrow$	$R_{\text{feas}} \uparrow$
0.5×	MPC	0.921	9.4	96.3
0.5×	GNN-RL	0.948	10.8	92.0
0.5×	TC-LEG	**0.892**	**7.6**	**96.0**
1.0×	MPC	0.861	6.9	99.1
1.0×	GNN-RL	0.872	6.8	95.9
1.0×	TC-LEG	**0.811**	**4.9**	**98.5**
1.5×	MPC	0.838	5.5	99.2
1.5×	GNN-RL	0.853	5.8	96.5
1.5×	TC-LEG	**0.790**	**4.1**	**98.6**
2.0×	MPC	0.831	5.1	99.2
2.0×	GNN-RL	0.850	5.4	96.2
2.0×	TC-LEG	**0.786**	**3.8**	**98.4**

cost and renewable curtailment because more renewable energy can be stored during high-generation periods and released during peak-load periods. For example, when the storage capacity increases from 0.5× to 1.5×, TC-LEG reduces the normalized operating cost from 0.892 to 0.790 and the renewable curtailment rate from 7.6% to 4.1%. This confirms the important role of hybrid energy storage in improving renewable energy accommodation and reducing grid purchasing cost.

However, the improvement becomes marginal when the storage capacity further increases from 1.5× to 2.0×. For TC-LEG, the operating cost decreases only from 0.790 to 0.786, and the curtailment rate decreases from 4.1% to 3.8%. This indicates that simply increasing storage capacity cannot indefinitely improve energy management performance. Once the main renewable surplus has been absorbed, additional storage provides limited benefit. This observation is useful for energy planning because it suggests that storage sizing should consider both economic benefit and marginal utilization efficiency.

TC-LEG consistently outperforms MPC and GNN-RL under all storage capacity settings. The advantage is especially clear under limited storage capacity, where the system must carefully decide whether to charge batteries, activate supercapacitors, curtail renewable energy, or purchase power from the grid. The proposed latent execution graph improves the coordination among these actions, enabling more efficient use of limited storage resources.

Sensitivity to prediction horizon. Table 7 analyzes the influence of the prediction horizon T_p. When $T_p = 3$, the model has limited look-ahead information and cannot fully anticipate future renewable fluctuations and load changes, resulting in a higher operating cost of 0.842. Increasing the horizon to 6 and 12 improves the cost and voltage performance because the control sequence generator can plan storage dispatch and curtailment decisions over a longer temporal window. The best operating cost is achieved when $T_p = 12$.

However, increasing the horizon to 24 does not further improve performance. Instead, the operating

Table 7. Sensitivity analysis under different prediction horizons on the IEEE 33-bus system

T_p	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{feas}} \uparrow$	$T_{\text{infer}} \downarrow$
3	0.842	1.3	98.2	11.8 ms
6	0.819	1.0	**98.7**	15.4 ms
12	**0.811**	**0.9**	98.5	18.6 ms
24	0.824	1.1	97.8	30.7 ms

cost increases to 0.824, and the action feasibility rate decreases to 97.8%. This is because longer-horizon sequence generation accumulates more uncertainty and increases the difficulty of maintaining physical feasibility over all future steps. In addition, the inference time increases from 18.6 ms at $T_p = 12$ to 30.7 ms at $T_p = 24$. Therefore, $T_p = 12$ provides a reasonable trade-off among economic performance, voltage security, feasibility, and computational efficiency. This result also suggests that energy-aware sequence generation should balance long-term planning capability with uncertainty accumulation and online dispatch requirements.

4.9. Ablation Study

To analyze the contribution of each component in TC-LEG, we conduct ablation experiments on the IEEE 33-bus system. The following variants are evaluated:

- **w/o Physical Graph**: The physical grid graph encoder is removed, and node features are directly fed into the sequence generator.
- **w/o Latent Execution Graph**: The latent execution graph is removed, and the generator relies only on physical grid representations.
- **w/o Topology Mask**: The topology-aware discrete action mask is removed.
- **w/o Digital Twin Verification**: The generated control sequence is directly executed without digital twin safety verification.
- **w/o Energy Loss**: The energy-aware cost loss is removed during training.
- **Full TC-LEG**: The complete proposed framework.

The results are reported in Table 8 and Fig. 5. Removing the physical grid graph increases the voltage violation rate from 0.9% to 2.6% and decreases the action feasibility rate from 98.5% to 94.1%. This confirms that physical topology information is essential for secure power grid operation, because voltage profiles and line loading conditions are strongly determined by the bus-branch structure.

Table 8. Ablation study of the proposed TC-LEG framework on the IEEE 33-bus system

Variant	$C_{\text{total}} \downarrow$	$R_{\text{volt}} \downarrow$	$R_{\text{curt}} \downarrow$	$R_{\text{feas}} \uparrow$
w/o Physical Graph	0.876	2.6	6.2	94.1
w/o Latent Execution Graph	0.849	1.5	5.7	97.4
w/o Topology Mask	0.838	3.2	5.5	90.8
w/o Digital Twin Verification	0.824	2.1	5.0	93.6
w/o Energy Loss	0.858	1.1	6.7	98.1
Full TC-LEG	**0.811**	**0.9**	**4.9**	**98.5**

Removing the latent execution graph increases the operating cost from 0.811 to 0.849 and the renewable curtailment rate from 4.9% to 5.7%. This shows that the latent execution graph contributes mainly to economic dispatch and renewable utilization by improving the coordination among heterogeneous actions. Without action-level dependency modeling, the model may still generate feasible actions, but the temporal cooperation among supercapacitor response, battery dispatch, renewable curtailment, and demand response becomes weaker.

Removing the topology mask causes the most severe drop in action feasibility, from 98.5% to 90.8%, and increases the voltage violation rate to 3.2%. This indicates that topology-aware action masking is critical for preventing invalid discrete actions, such as infeasible switching or storage operations at inappropriate nodes. Removing digital twin verification also reduces feasibility and increases voltage violations, demonstrating that post-generation safety checking is necessary because some violations can only be detected through power flow simulation. Finally, removing the energy-aware loss increases the operating cost and renewable curtailment rate, confirming that the proposed training objective helps the model optimize energy-related goals rather than merely imitating expert actions.

Overall, the ablation results show that each component contributes to TC-LEG from a different perspective: physical topology modeling improves security, latent execution graph learning improves coordination, action masking improves feasibility, digital twin verification improves safety, and energy-aware loss improves economic and renewable-energy performance.

5. Conclusion

This paper proposed TC-LEG, a topology-constrained latent execution graph learning framework for sequential energy management in digital-twin smart grids. The physical grid graph represents electrical connectivity and operating states, whereas the latent execution graph represents directed conditional dependencies among action tokens. TC-LEG combines these representations with a hybrid discrete–continuous sequence generator and a closed-loop digital-twin verification

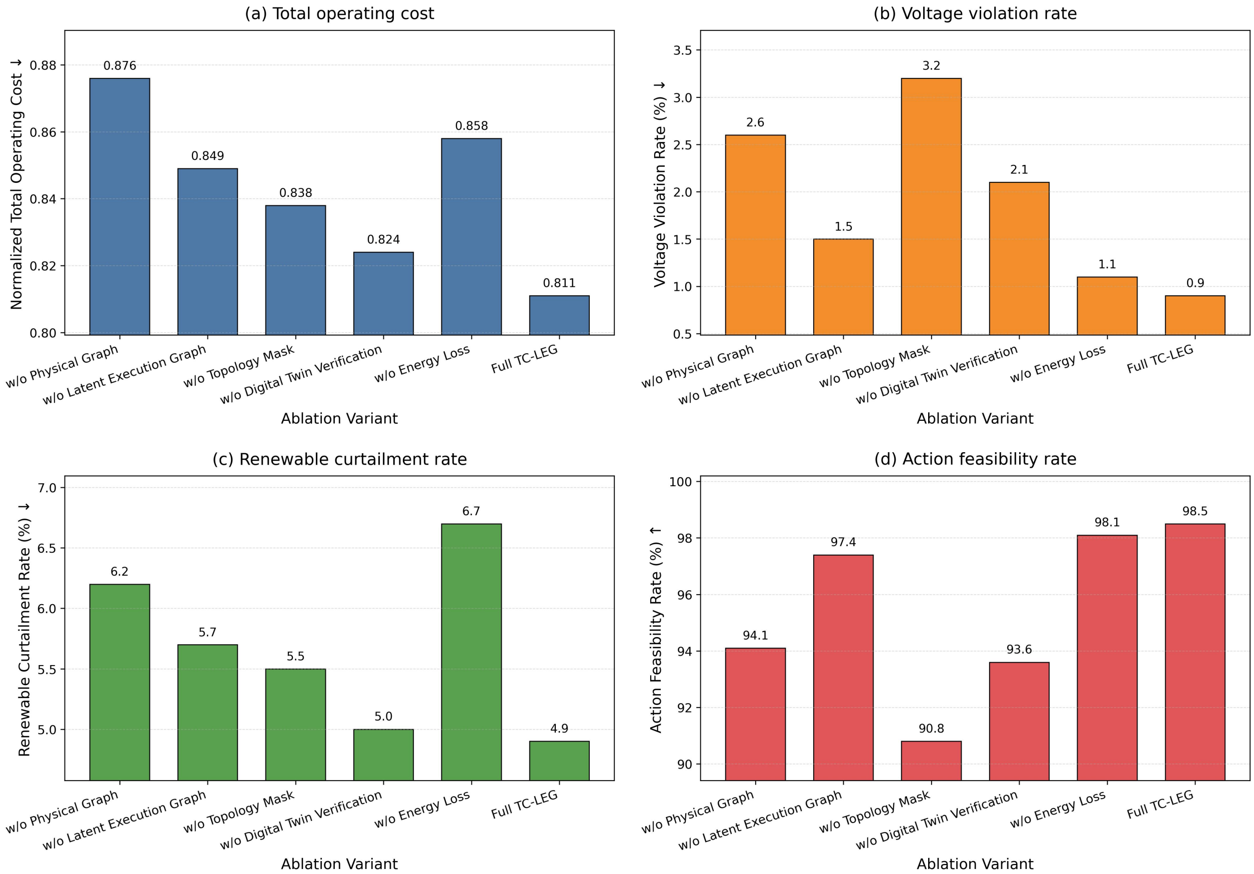

Figure 5. Ablation study of the proposed TC-LEG framework on the IEEE 33-bus system. Four evaluation metrics are reported: normalized total operating cost, voltage violation rate, renewable curtailment rate, and action feasibility rate. The results show that physical topology modeling, latent execution graph learning, topology-aware action masking, digital twin verification, and energy-aware optimization are all necessary for achieving secure, feasible, and cost-efficient smart grid energy management

mechanism. The verification module synchronizes the current virtual state, checks each candidate action, projects infeasible candidates onto a state-dependent feasible set, advances the virtual state using the accepted action, and feeds the accepted action back to subsequent decoding.

Experiments on the IEEE 33-bus and IEEE 69-bus systems show that TC-LEG reduces normalized operating cost, voltage violations, and renewable curtailment while maintaining high action feasibility. On the IEEE 33-bus system, it achieves a normalized operating cost of 0.811, a voltage violation rate of 0.9%, a renewable curtailment rate of 4.9%, and an action feasibility rate of 98.5%. Its inference latency remains far below the 15-minute control interval used in the experiments.

Several limitations should be acknowledged. First, model performance depends on the coverage and quality of expert trajectories obtained from historical records, optimization solvers, or digital-twin simulations. Second, the current device and network models simplify some dynamic, protection, and communication effects, and errors in the digital-twin model may propagate to verification and correction decisions. Third, the experiments use benchmark distribution systems and therefore do not fully reproduce the scale, data quality, organizational constraints, and heterogeneous equipment found in real distribution networks. Finally, the current digital twin uses interval-based state synchronization and offline-calibrated parameters rather than continuous online model calibration. Future work will investigate uncertainty-aware training, online twin calibration, more detailed device and protection models, communication delay and measurement noise, and validation on utility-scale distribution networks.

References

[1] Glass P, Di Marzo Serugendo G. Coordination Model and Digital Twins for Managing Energy Consumption and Production in a Smart Grid. Energies. 2023;16(22):7629.

[2] Asadi S, Naeini HK, Hassanlou D, Pishahang A, Najafabadi SA, Sharifi A, et al. AI-Powered Digital Twin Frameworks for Smart Grid Optimization and Real-Time Energy Management in Smart Buildings: A Survey. Computer Modeling in Engineering & Sciences. 2025;145(2):1259-301.

[3] Stogia M, Dimara A, Papaioannou C, Eleftheriou O, Papaioannou A, Krinidis S, et al. ENACT: Energy-Aware, Actionable Twin Utilizing Prescriptive Techniques in Home Appliances. Smart Cities. 2025;8(5):155.

[4] Aghazadeh Ardebili A, Zappatore M, Ramadan AIHA, Longo A, Ficarella A. Digital Twins of Smart Energy Systems: A Systematic Literature Review on Enablers, Design, Management and Computational Challenges. Energy Informatics. 2024;7(1):94.

[5] Uvarajan K. Secure Digital-Twin-Assisted Multi-Agent Learning Architectures for Reliable Energy-Oriented Decision-Making in Distributed Vehicular Services. Transactions on Internet Security, Cloud Services, and Distributed Applications. 2025:1-6.

[6] Reddy BNK, Krishna YC. An Explainable, Digital-Twin-Assisted Paradigm for Self-Reconfiguring and Energy-Resilient Autonomous Transportation. IEEE Transactions on Consumer Electronics. 2026;72(2):4872-82.

[7] Pei L, Xu C, Yin X, Zhang J. Multi-Agent Deep Reinforcement Learning for Cloud-Based Digital Twins in Power Grid Management. Journal of Cloud Computing. 2024;13(1):152.

[8] Alghamdi M, Abadleh A, Mnasri S, Alrashidi M, Alkhazi IS, Alghamdi A, et al. Hazard- and Fairness-Aware Evacuation with Grid-Interactive Energy Management: A Digital-Twin Controller for Life Safety and Sustainability. Sustainability. 2026;18(1):133.

[9] Hassouna M, Holzhüter C, Lytaev P, Thomas J, Sick B, Scholz C. Graph Reinforcement Learning in Power Grids: A Survey. arXiv preprint arXiv:240704522. 2024.

[10] Li H, Pangborn HC, Kovalenko I. Hierarchical Model Predictive Control for Energy-Aware Scheduling of Digital Twin-Based Batch Manufacturing Systems. IEEE Transactions on Automation Science and Engineering. 2026;23:291-305.

[11] Cao Z, Wang R, Zhou X, Tan R, Wen Y, Yan Y, et al. Adaptive capacity provisioning for carbon-aware data centers: a digital twin-based approach. IEEE Transactions on Sustainable Computing. 2025;10(4):730-43.

[12] Wu S, Xie R, Tang Q, Zhao H, Chen W, Ranaweera P, et al. Digital Twin-Driven Dual-Time-Scale Orchestration for Synergistic Computing Power and Electric Power Systems. IEEE Transactions on Green Communications and Networking. 2026;10:2789-803.

[13] Nie Z, Tian H, Yin Y, Zhou Y, Li W, Xiong Y, et al. System-Level Optimization of AUV Swarm Control and Perception: An Energy-Aware Federated Meta-Transfer Learning Framework with Digital Twin Validation. Journal of Marine Science and Engineering. 2026;14(4):384.

[14] Vamvakas D, Papaioannou I, Tsaknakis C, Sgouros T, Korkas C. Generative AI for Sustainable Smart Environments: A Review of Energy Systems, Buildings, and User-Centric Decision-Making. Energies. 2025;18(23):6163.

[15] Mohamed EM, Fouda MM. A Digital Twin-Driven Combinatorial Neural Bandit Approach for Multi-Hop UAV Route Optimization in Delay-Doppler-Aware FANETs. IEEE Internet of Things Journal. 2026. In press.

[16] Bysko S, Bysko S, Blachowicz T. Virtual Commissioning and Digital Twins for Energy-Aware Industrial Electric Drive Systems. Energies. 2025;18(20):5375.

[17] Hu J, Shan Y, Yang Y, Parisio A, Li Y, Amjady N, et al. Economic Model Predictive Control for Microgrid Optimization: A Review. IEEE Transactions on Smart Grid. 2024;15(1):472-84.

[18] Guo C, Wang X, Zheng Y, Zhang F. Real-Time Optimal Energy Management of Microgrid with Uncertainties Based on Deep Reinforcement Learning. Energy. 2022;238:121873.

[19] Wang Y, Qiu D, Sun M, Strbac G, Gao Z. Secure Energy Management of Multi-Energy Microgrid: A Physical-Informed Safe Reinforcement Learning Approach. Applied Energy. 2023;335:120759.

[20] Chen L, Gu Q, Jiang K, Zhao L. A3C-based and dependency-aware computation offloading and service caching in digital twin edge networks. IEEE Access. 2023;11:57564-73.

[21] Li D, Han D, Crespi N, Minerva R, Raza SM, Farahbakhsh R, et al. Blockchain in the Digital Twin Context: A Comprehensive Survey. ACM Computing Surveys. 2025;58(6):1-35.

[22] Yigit Y, Maglaras LA, Buchanan WJ, Canberk B, Shin H, Duong TQ. AI-enhanced digital twin framework for cyber-resilient 6G Internet of Vehicles networks. IEEE Internet of Things Journal. 2024;11(22):36168-81.

[23] Dong Z, Zhang Z, Li Z, Li X, Qin J, Liang C, et al. A Survey of Battery–Supercapacitor Hybrid Energy Storage Systems: Concept, Topology, Control and Application. Symmetry. 2022;14(6):1085.

[24] Kabir T. Digital Twin–Enabled Optimization of Electrical, Instrumentation, And Control Architectures In Smart Manufacturing And Utility-Scale Systems. International Journal of Scientific Interdisciplinary Research. 2025;6(1):404-51.

[25] Fu Y, Turkcan MK, Ghasemi M, Mo Z, Zang C, Adhikari A, et al. AI-Powered CPS-Enabled Vulnerable-User-Aware Urban Transportation Digital Twin: Methods and Applications. IEEE Transactions on Intelligent Transportation Systems. 2026;27(5):4949-66.

[26] Georgiadis GP, Dimitriadis CN, Georgiadis MC. Decarbonizing the industry sector: current status and future opportunities of energy-aware production scheduling. Processes. 2025;13(6):1941.

[27] Mamodiya U, Kishor I, Mudholkar P, Alqutaish A, Alradwan G, Obeidat M. A Robust Smart Grid-Aware Cloud Computing Framework for Sustainable Energy Management. International Journal of Advances in Soft Computing and its Applications. 2026;18(1):396-433.

[28] Verma PK, Vasudev H. AI and Data-Driven Sustainable Design and Manufacturing. In: Smart Sustainable Materials Advancing Remanufacturing and Technology: Sustainable Technology for Product Design, Manufacturing, and Remanufacturing: An Emerging Road Map from Industry 4.0 to 6.0 (Volume II). Springer; 2026. p. 347-67.

[29] Liao W, Bak-Jensen B, Pillai JR, Wang Y, Wang Y. A review of graph neural networks and their applications in power systems. Journal of Modern Power Systems and Clean Energy. 2021;10(2):345-60.

[30] Harrold DJB, Cao J, Fan Z. Renewable Energy Integration and Microgrid Energy Trading Using Multi-Agent Deep Reinforcement Learning. Applied Energy. 2022;318:119151.

[31] Aljumaily M, Abed H. AI-Driven Digital Twin Frameworks for Predictive Monitoring of IoT Networks in Harsh Environments. International Journal of Mechatronics, Robotics, and Artificial Intelligence. 2026;2(1):1-9.