EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

DT-HERMES: Digital-Twin-Guided Emergency Workflow Modeling and Hierarchical Task Decomposition for Power Grid Cybersecurity Response with LLM-MAS Hybrid Agents

Yun Su[1], Xuejun Xiong[1], Mengyuan Zhang[1], Hongshan Yang[2*]

[1]State Grid Shanghai Municipal Electric Power Company, Shanghai, 200437, China
[2]Transwarp Technology (Shanghai) Co., Ltd, Shanghai, 200233, China

Abstract

INTRODUCTION: Power grid cybersecurity incidents are no longer confined to isolated information systems, but increasingly propagate across cyber and physical layers, where malicious intrusion, abnormal communication, and dispatching disturbances may interact with grid operation states in real time. This coupling makes conventional emergency response approaches insufficient, because predefined plans are often difficult to adapt to evolving attack contexts, changing system states, and the coordination needs of multiple operational roles.
OBJECTIVE: To address this problem, this paper proposes DT-HERMES(Digital-Twin-Guided Emergency Workflow Modeling and Hierarchical Task Decomposition for Power Grid Cybersecurity Response), a digital-twin-enabled emergency workflow modeling and hierarchical task decomposition method based on large language model (LLM)-multi-agent system (MAS) hybrid agents for power grid cybersecurity response. DT-HERMES integrates cyber-physical digital twin models with hybrid intelligent agents to construct a unified response space linking security alerts, grid operating conditions, and emergency actions.
METHODS: In the proposed framework, large language model (LLM)-based agents undertake semantic understanding, knowledge reasoning, and workflow generation, while specialized agents are introduced for topology-aware verification, state-estimation alignment, safety-boundary checking, and cross-role task coordination. On this basis, a hierarchical task decomposition mechanism is designed to translate high-level response objectives into executable subtasks with explicit temporal order, responsibility assignment, and operational constraints.
CONCLUSION: Different from purely language-driven response generation methods, DT-HERMES leverages the digital twin environment to validate action feasibility, assess potential system impact, and iteratively revise the disposal process according to real-time monitoring feedback. In this way, the proposed method supports a transition from static plan matching to adaptive, verifiable, and closed-loop emergency orchestration. The study provides a digital-twin-oriented intelligent response paradigm for resilient smart energy systems and offers practical support for real-time monitoring, state estimation, and coordinated control under cyberattack scenarios.

Received on 19 August 2026; accepted on 11 October 2026; published on 24 November 2026

Keywords: Power grid cybersecurity, digital twin, emergency response, LLM-MAS hybrid agents

Copyright © 2026 Yun Su *et al.*, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi:10.4108/ew.14604

1. Introduction

As power grids continue to evolve toward deep digitalization and pervasive interconnection, the operational

*Corresponding author. Email: hongshan.yang@transwarp.io

boundary of the energy system is expanding from physical infrastructure to a tightly coupled cyber-physical space[1, 2]. In this environment, cybersecurity incidents are no longer confined to isolated information assets. Instead, they may propagate through communication networks, control services, dispatching platforms, and sensing links, eventually influencing state perception, operational decision-making, and even physical control behavior[1, 3]. For smart energy systems, such cross-layer disturbance has made cybersecurity emergency response a critical part of resilient grid operation rather than a peripheral support function.

Conventional emergency response strategies in power grid cybersecurity are still largely built upon predefined plans, expert rules, and manually organized workflow templates[2, 4]. Although such approaches remain useful in relatively structured scenarios, their limitations become evident when facing multi-source alarms, evolving attack chains, and rapidly changing grid states. A response plan that is logically complete at the managerial level may still be difficult to execute under actual operating constraints. In particular, existing methods often lack explicit coordination between incident interpretation, physical state awareness, topology dependency, and control feasibility[5, 6]. As a result, emergency actions may be delayed, overly conservative, or insufficiently aligned with the real-time condition of the grid.

The technical motivation for hierarchical task decomposition in this setting is that a power-grid cyber incident is not a single planning problem with a fixed state transition graph. Conventional workflow templates usually assume a predetermined execution path, hierarchical task networks mainly require hand-designed methods and operators, and planning graphs are effective only when the state predicates and action preconditions are sufficiently complete. In coupled cyber-physical emergencies, however, the available evidence, control reachability, operating margins, and role permissions may change after each response step. Therefore, the response objective must be decomposed into subtasks whose dependencies, permissions, safety constraints, and feedback triggers can be checked and revised online. This is the reason why DT-HERMES adopts hierarchical task decomposition under digital-twin constraints rather than using a conventional static workflow, a pure HTN planner, or a planning-graph formulation alone.

Recent advances in large language models have opened a new possibility for intelligent emergency response. Their capabilities in semantic understanding, knowledge extraction, contextual reasoning, and task planning make them attractive for processing heterogeneous alarm descriptions, operating manuals, and historical response cases[7]. However, directly applying a general-purpose language model to power grid emergency response is not sufficient. Language-driven generation alone cannot guarantee that the produced workflows satisfy operational safety boundaries, state estimation consistency, or topology-aware execution constraints[8, 9]. In addition, cybersecurity response in power systems is inherently collaborative, involving security monitoring personnel, dispatching operators, communication engineers, field responders, and supervisory decision-makers. A single monolithic reasoning model is therefore ill-suited to manage the full chain from situation interpretation to action coordination[10].

Digital twin technology provides an effective foundation for addressing these limitations. In smart energy systems, a digital twin is not merely a visual replica of physical assets, but a unified environment that connects real-time monitoring, state estimation, mechanism-aware analysis, operational validation, and control feedback[11]. Embedding cybersecurity emergency response into such a digital twin environment enables response processes to be evaluated against evolving system conditions rather than generated only from static knowledge. This creates the possibility of transforming emergency disposal from text-level recommendation into a closed-loop procedure that can be interpreted, validated, adjusted, and coordinated in accordance with cyber-physical system dynamics.

Motivated by this need, this paper proposes DT-HERMES(Digital-Twin-Guided Emergency Workflow Modeling and Hierarchical Task Decomposition for Power Grid Cybersecurity Response) for emergency workflow modeling and hierarchical task decomposition in power grid cybersecurity response. The core idea is to treat emergency disposal as a constrained collaborative process in which multi-role intelligent agents interact with a cyber-physical digital twin to continuously refine response decisions. Within this framework, large language model (LLM)-based agents are responsible for alarm understanding, intent extraction, and initial workflow construction, while specialized agents perform state-consistency checking, topology-aware verification, safety-boundary assessment, and cross-role coordination. Through this hybrid mechanism, high-level emergency objectives can be progressively decomposed into executable subtasks with explicit temporal logic, operational dependencies, and responsibility assignment. More importantly, the resulting workflows are not accepted solely on the basis of semantic plausibility, but are further evaluated and revised through the digital twin according to monitoring feedback and system-level feasibility.

Compared with existing studies, the value of DT-HERMES lies in its ability to bridge cybersecurity incident handling with real-time grid operation requirements in a unified intelligent framework. It does not simply generate emergency procedures, but organizes

response reasoning, task decomposition, and disposal verification into a closed-loop process that is more adaptive to dynamic attack scenarios and more compatible with the real operating characteristics of smart energy systems.

More specifically, the novelty of DT-HERMES is reflected in four concrete technical designs. First, it defines an event-window-level cyber-physical state representation that aligns alarm semantics, operating-state vectors, and response-action records in one indexed decision object. Second, it realizes a hybrid mapping mechanism that combines rule-based alarm normalization, knowledge-graph entity linking, LLM-based semantic extraction, and digital-twin state projection. Third, it implements a constraint-aware multi-agent aggregation algorithm rather than a sequential dialogue among different LLM roles. Fourth, it evaluates the generated response with separated targets for attack recognition and response-objective generation, so that the reported results can be traced to specific technical functions rather than only to the overall architecture.

The main contributions of this paper are summarized as follows:

- A digital-twin-guided emergency response framework is developed for power grid cybersecurity scenarios, in which cyber alerts, physical operating states, and disposal actions are integrated into a unified cross-layer decision space. The framework specifies the data construction procedure for event-window samples and separates attack-recognition targets from response-objective-generation targets.

- A hybrid multi-agent mechanism is designed in DT-HERMES to combine the semantic reasoning capability of LLM-based agents with specialized verification agents for topology constraints, state estimation alignment, safety boundaries, and collaborative execution. A concrete constraint-aware aggregation function is introduced to resolve conflicts according to feasibility score, role permission, priority, and safety veto rules.

- A hierarchical task decomposition method is proposed to transform high-level emergency objectives into executable subtasks with explicit temporal ordering, role assignment, and operational constraints, thereby improving the consistency and practicality of emergency workflows. Compared with conventional workflows, HTNs, and planning graphs, the proposed decomposition explicitly supports online revision when the digital twin detects state deviation or constraint violation.

- A closed-loop response paradigm is established by coupling workflow generation with digital twin validation and real-time monitoring feedback, enabling adaptive revision of emergency actions under evolving cyberattack conditions. The study further reports ablation, robustness, and parameter sensitivity results to support the contribution of each mechanism.

2. Related Work

Research on power grid cybersecurity emergency response has mainly evolved along three related directions, namely emergency workflow modeling, digital-twin-based situation representation, and intelligent collaborative decision-making. These directions respectively address how response procedures are organized, how system states are mapped and interpreted, and how emergency tasks are coordinated. However, most existing studies still focus on one of these aspects in isolation, and relatively few works provide a unified closed-loop mechanism that jointly considers real-time monitoring, operational constraints, task decomposition, and dynamic revision.

In emergency workflow modeling, a common line of work relies on plan libraries, expert rules, and predefined response templates. A typical implementation first classifies incidents according to alarm type, affected assets, or severity level, and then activates a corresponding response path composed of stages such as alarm confirmation, impact assessment, isolation control, service recovery, and post-event reporting. Some studies describe event evolution through attack trees, fault trees, or event trees[12, 13], while others formalize task sequencing and execution states using Petri nets, BPMN-like process models, or finite-state representations[14, 15]. These methods offer clear procedural structure and strong traceability, which makes them suitable for integration with existing utility operation practices. Nevertheless, they are usually built on relatively static assumptions about event evolution. In actual cybersecurity incidents, attack chains may unfold across multiple stages, available observations may be incomplete, and response priorities may shift with changing grid conditions. Under such circumstances, fixed workflow models often struggle to maintain both flexibility and operational relevance.

To improve adaptability, several studies have introduced knowledge graphs, case-based reasoning, and rule-driven inference into emergency response modeling. Their typical idea is to organize the relationships among attack behaviors, target assets, service functions, risk consequences, and response actions into a semantic association network[3, 16], and then infer the likely impact scope or candidate actions from

incoming alarms. Some methods further reuse historical tickets, operation logs, or expert experiences to retrieve similar cases and generate response suggestions accordingly[17, 18]. Compared with purely template-driven mechanisms, these approaches provide richer contextual reasoning and are better suited to heterogeneous or partially unstructured alarm descriptions. However, their limitations remain evident. Knowledge-graph-based approaches are highly dependent on the completeness and maintenance quality of prior knowledge, which makes them vulnerable when facing emerging attack patterns or complex cross-domain disturbance propagation. Case-based reasoning can reuse valuable experience, but its effectiveness is strongly constrained by the representativeness of historical samples. More importantly, these methods often stop at recommending response strategies, while offering limited support for systematically decomposing a high-level response objective into executable, verifiable, and role-specific subtasks.

Another important stream of research is digital twin technology for smart energy systems. Existing work in this area has increasingly moved from single-device virtual mapping to multi-layer twin environments covering equipment states, communication links, control logic, and operational topology. In such settings, physical measurements, network conditions, dispatching commands, and system configuration can be integrated into a unified virtual space[19, 20] for real-time monitoring, anomaly detection, and consequence analysis. Some methods identify abnormal propagation by comparing observed states, estimated states, and predicted states, for example by jointly analyzing state-estimation residuals, communication delay variations, and telemetry credibility changes. Other studies simulate the impact of false data injection, communication interruptions, or local controller malfunction on power flow distribution, regional stability, and recovery behavior[21, 22]. These efforts demonstrate that digital twins can provide a much stronger basis for cyber-physical situation awareness than conventional offline analysis.

Despite these advances, most digital-twin-oriented studies emphasize state visibility and consequence prediction rather than the organization of emergency response tasks[23]. In other words, they are effective at describing what is happening in the system and what might happen next, but are less explicit about how the response process itself should be generated, coordinated, and revised. Even when disposal actions are included in the twin environment, they are often limited to a small number of predefined operations, such as isolating nodes, switching links, recovering channels, or enabling redundant services. This is useful when the action space is narrow and well-structured, but it is less adequate for complex emergency scenarios involving multi-role communication, procedural authorization, human verification, and iterative replanning. As a result, the digital twin often serves as a high-fidelity monitoring and simulation platform, yet does not fully function as an active engine for workflow construction and task orchestration.

In intelligent decision-making, one line of work follows the traditional multi-agent paradigm. Different operational units, control nodes, communication modules, or service functions are modeled as autonomous agents that interact through negotiation, hierarchical control, distributed optimization, or consensus-based coordination[24]. Such approaches have been widely used in fault restoration, distributed energy management, and communication-resource coordination. Their strength lies in handling structured problems with explicit state variables and well-defined interaction rules. However, when transferred to cybersecurity emergency response, they often show limited capacity in interpreting heterogeneous alarm semantics, extracting latent incident intent, or composing complex response workflows from partially structured information.

A more recent line of research employs large language models to enhance intelligent coordination. In these methods, different agents may assume roles such as incident interpreter, planner, tool executor, reviewer, or memory manager, so that complex tasks can be solved through role-specialized collaboration. In cybersecurity applications, similar mechanisms have already been explored for alarm explanation, attack-chain analysis, response recommendation, and automated ticket generation. Their main advantage is the ability to process unstructured text, integrate cross-source information, and produce flexible reasoning outputs under incomplete context. This makes them especially attractive for emergency response scenarios where procedural knowledge, operational manuals, and alarm messages coexist in heterogeneous forms. Yet directly applying such methods to power grid cybersecurity remains problematic. The generated workflows may be semantically plausible while still violating state-estimation consistency, topology dependencies, communication reachability, or safety boundaries in actual grid operation. Moreover, even if multiple agents can collaborate effectively at the textual level, their outputs may still conflict at the system level when no shared physical-state anchor is imposed. Existing LLM-based agent frameworks also tend to rely on textual feedback or tool-return messages for refinement, which is still far from the continuously evolving and strongly coupled feedback environment required in cyber-physical power systems[20].

It is also necessary to distinguish genuine multi-agent collaboration from sequential interaction among multiple LLM roles. In a sequential role-prompting

process, one role usually produces an output and another role comments on or rewrites it, while all roles implicitly share the same unrestricted textual context. By contrast, a collaborative multi-agent response system should maintain distributed state representations, define role-specific observation scopes and action permissions, and provide an explicit conflict-resolution mechanism when different agents recommend incompatible actions. In DT-HERMES, the alarm interpretation agent, topology verification agent, state-estimation agent, safety-boundary agent, and coordination agent do not simply speak in turn. Each agent receives a different local observation $o_i^{(t)}$, accesses a constrained memory $m_i^{(t)}$, operates under a role permission set $\mathcal{P}_i$, and submits a decision candidate that is fused by the aggregation function $\Psi(\cdot)$. This design turns multi-agent reasoning from a sequential prompting style into a constrained distributed decision process.

Overall, prior studies have laid important foundations in workflow representation, state mapping, and intelligent coordination, but several gaps remain. Existing approaches generally lack a unified framework that connects cybersecurity alarms, physical operating states, and emergency actions in a coherent decision space. They also tend to favor either rigid rule enforcement or flexible language generation, without adequately combining semantic intelligence with engineering feasibility constraints. In addition, many studies describe emergency response only at a coarse procedural level, while offering insufficient support for decomposing high-level response goals into executable subtasks with explicit temporal dependencies, role assignments, and safety boundaries. These limitations motivate the development of DT-HERMES, which aims to integrate digital twin validation, hybrid multi-agent reasoning, and hierarchical task decomposition into a closed-loop response mechanism for power grid cybersecurity emergencies.

3. Methodology

DT-HERMES is developed for power grid cybersecurity emergency response under digital twin constraints[26], with the objective of converting unstructured security alarms, dynamic grid operating states, and multi-role emergency actions into a closed-loop response process that is modelable, decomposable, and continuously revisable. In contrast to conventional methods that either rely on predefined plans or directly generate response suggestions from a single reasoning engine, DT-HERMES treats emergency disposal as a constrained collaborative decision problem involving semantic interpretation, cyber-physical state alignment, workflow generation, constraint verification, hierarchical task decomposition, and feedback-driven replanning. As illustrated in Fig. 1 and detailed in Algorithm 1, the overall framework consists of a digital-twin-guided workflow modeling module, a hybrid agent collaborative reasoning module, and a hierarchical task decomposition module. These three modules are not independent components, but are connected through an iterative loop in which semantic understanding is grounded by grid states, candidate workflows are verified against operational constraints, and the decomposed tasks are further updated according to real-time monitoring feedback. To provide a consistent decision basis for subsequent modeling and reasoning, the closed-loop emergency state of the power grid is first represented in a unified form. Let the integrated response state at time t be defined as

$$\mathcal{X}_t = (\mathcal{A}_t, \mathcal{G}_t, \mathcal{R}_t) \tag{1}$$

where $\mathcal{A}_t$ denotes the set of cybersecurity alarms at time t, including attack categories, abnormal logs, alarm confidence, and affected cyber assets; $\mathcal{G}_t$ denotes the grid operating state at time t, including topology structure, state estimation results, critical equipment conditions, measurement credibility, and communication link status; and $\mathcal{R}_t$ denotes the set of generated or executed emergency response actions at time t. The purpose of introducing $\mathcal{X}_t$ is to place security observations, physical operating conditions, and response actions into the same decision state, so that workflow generation is no longer driven only by textual alarms or only by numerical operating variables. In this representation, $\mathcal{X}_t$ serves as the global emergency state, while $\mathcal{A}_t$, $\mathcal{G}_t$, and $\mathcal{R}_t$ correspond to the cyber situation, physical situation, and response situation, respectively. The advantage of this unified formulation is that it explicitly preserves the coupling between cyber disturbances and physical operation, which is essential for emergency response scenarios where the same cyber event may induce different disposal priorities under different grid conditions.

In the implementation, the time index t denotes an event window rather than an isolated log line. For network-flow datasets, t corresponds to a 60 s window with a 30 s stride; for process-control datasets, it corresponds to a 1 s sampled process state; for EPIC, it corresponds to one synchronized cyber-physical experiment record. The alarm set is represented as $\mathcal{A}_t = \{(l_j, e_j, s_j, c_j, b_j)\}_{j=1}^{n_t}$, where l_j is the normalized attack label, $e_j \in \mathbb{R}^{d_e}$ is the text embedding of the alarm description, s_j is the affected service or asset identifier, $c_j \in [0, 1]$ is the alarm confidence, and b_j is the event timestamp within the window. The grid state is represented as $\mathcal{G}_t = (\mathbf{x}_t, \mathbf{H}_t, \mathbf{Q}_t, \mathbf{L}_t)$, where

Table 1. Cross-cutting comparison of related studies and DT-HERMES

Research stream	Workflow representation	Digital-twin or state grounding	Decision coordination	Main limitation for emergency response
Template/rule workflow models[12–15]	Predefined stages, event trees, Petri nets, or finite-state workflows	Usually weak or offline	Mainly manual or rule-triggered	Limited adaptability to changing attack chains and operating states
Knowledge-graph and case-based response[16–18, 25]	Semantic association among attacks, assets, impacts, and actions	Partial state grounding through asset or impact links	Rule inference or case retrieval	Insufficient support for executable subtask decomposition and feedback revision
Digital-twin-based power-system analysis[19–22]	Usually action lists or simulation scenarios	Strong monitoring, estimation, and consequence simulation	Limited workflow-level orchestration	Twin mainly supports visibility and prediction, not active task construction
LLM-based role agents[10]	Flexible natural-language plans and iterative feedback	Typically text/tool-feedback based	Often sequential role interaction	May generate semantically plausible but physically infeasible workflows
DT-HERMES	Graph-structured hierarchical subtasks with dependencies and constraints	Cyber-physical digital twin with state estimation, topology, and safety margins	Distributed role agents with explicit permissions and constraint-aware aggregation	Designed to close the above gap by jointly modeling alarms, states, actions, and revision

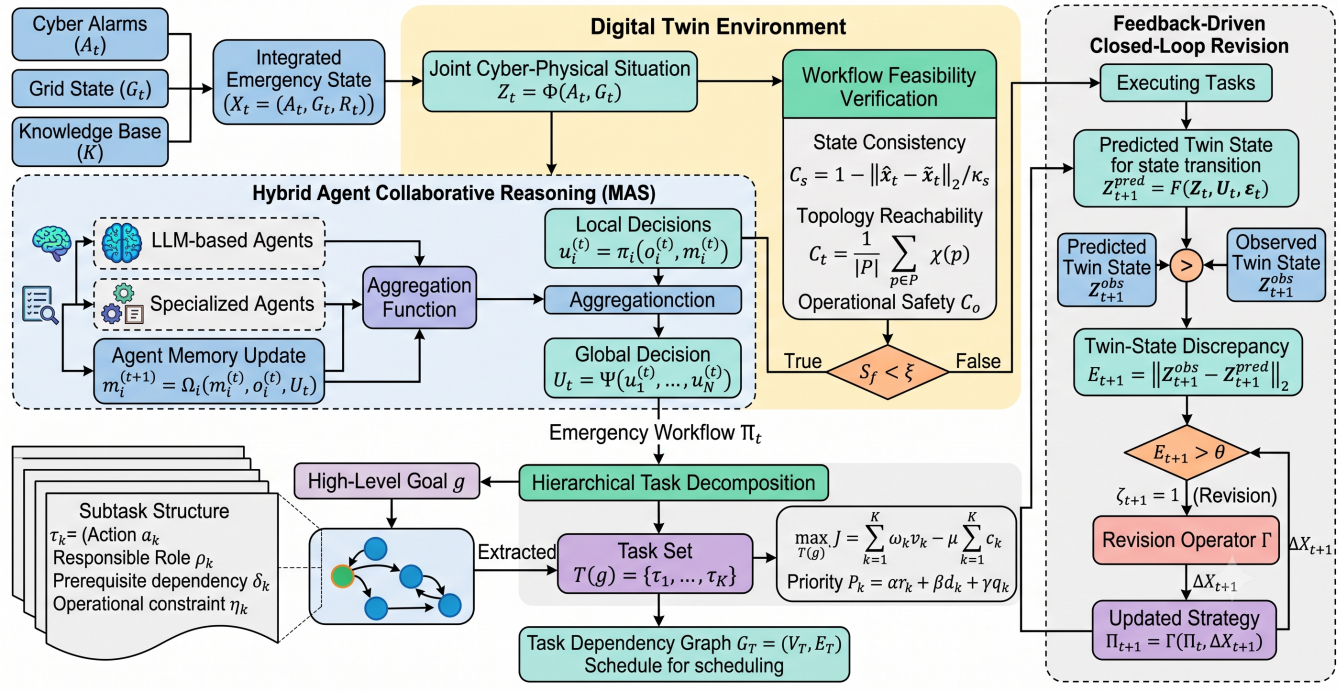

Figure 1. The closed-loop algorithm diagram of DT-HERMES for power grid cybersecurity response

$\mathbf{x}_t \in \mathbb{R}^{d_g}$ contains normalized measurements or state-estimation variables, $\mathbf{H}_t \in \{0,1\}^{n_b \times n_b}$ is the bus/device or process-topology adjacency matrix, $\mathbf{Q}_t \in [0,1]^{d_g}$ stores measurement credibility scores, and $\mathbf{L}_t \in \{0,1\}^{n_l}$

records communication-link availability. The response set is represented as $\mathcal{R}_t = \{(a_h, \rho_h, \sigma_h, \Delta z_h)\}_{h=1}^{m_t}$, where a_h is the action type, ρ_h is the responsible role, $\sigma_h \in \{\text{planned}, \text{executing}, \text{finished}, \text{failed}\}$ is the execution state, and Δz_h is the expected twin-state change. Missing feature fields are encoded by a binary mask and are not treated as zero-valued observations.

Based on the unified emergency state, DT-HERMES uses a digital twin environment to perform cross-layer situation mapping. The goal is not only to reflect current grid status in a virtual space, but also to transform alarm semantics into operation-oriented impact representations that can directly support response modeling. The mapping from alarms and operating states to the digital twin situation representation is defined as

$$\mathcal{Z}_t = \Phi(\mathcal{A}_t, \mathcal{G}_t) \tag{2}$$

where $\mathcal{Z}_t$ denotes the joint cyber-physical situation representation in the digital twin, and $\Phi(\cdot)$ denotes the mapping operator from the alarm space and the operating-state space to the twin feature space. The purpose of Eq. (2) is to translate semantic attack descriptions into physically interpretable consequences, such as which measurements may become unreliable, which communication paths may be degraded, and which control actions may face increased risk. Here, $\mathcal{Z}_t$ is not a simple visualization output, but a mechanism-aware intermediate state used by the following reasoning and decomposition modules. The advantage of this mapping lies in its ability to bridge security interpretation and power system operation analysis, making it possible to reason over emergency actions with awareness of physical feasibility rather than text-level plausibility alone.

The mapping $\Phi(\cdot)$ is realized as an integrated rule–knowledge–LLM–twin projection procedure. First, expert rules normalize raw labels and logs into attack tactics, affected services, and candidate response objectives. Second, a domain knowledge graph links entities such as substations, meters, communication channels, controllers, sensors, actuators, and protection functions to their cyber and physical dependencies. Third, an LLM-based extractor converts unstructured alarm text and operating notes into typed tuples $(\text{attack}, \text{asset}, \text{symptom}, \text{impact})$. Fourth, the typed tuples are projected to the digital-twin state through asset-to-topology matching and measurement-channel matching. Thus, Φ is not a purely learned black-box function; it is a hybrid mapping whose symbolic rules provide deterministic normalization, whose knowledge graph provides dependency relations, and whose LLM component handles unstructured semantics.

To further characterize the evolution of the digital twin state under emergency actions, the state transition in the twin environment is modeled as

$$\mathcal{Z}_{t+1} = F(\mathcal{Z}_t, U_t, \varepsilon_t) \tag{3}$$

where $F(\cdot)$ denotes the twin-state transition operator, U_t denotes the coordinated response action generated at time t, and ε_t denotes external disturbances or uncertainty factors, such as delayed observations, incomplete logs, communication fluctuations, or unknown attack progression. The purpose of introducing Eq. (3) is to represent the fact that emergency response is not a one-shot decision but a dynamic intervention process whose effects unfold over time. In this formulation, $F(\cdot)$ can be understood as a mechanism-aware evolution function that reflects how cyber-physical system states change under both ongoing incidents and applied countermeasures. The advantage of this formulation is that it enables DT-HERMES to assess not only whether a candidate workflow is feasible at the current moment, but also whether it is likely to remain effective after the system state evolves.

In this work, $F(\cdot)$ is connected to concrete simulators according to the dataset type. For EPIC, F is instantiated by a cyber-physical power-system twin containing a network-topology model, telemetry credibility model, and power-flow/state-estimation update model. For SWaT and WaDi, F is implemented as a process-control twin that updates sensor and actuator states according to recorded process dynamics and attack-stage transitions. For CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021, where physical grid variables are not fully available, F is instantiated as a network/IIoT state transition model over communication reachability, service availability, attack-stage probability, and response-objective status. In all cases, F receives candidate actions such as alarm confirmation, link isolation, channel switching, controller verification, measurement trust adjustment, and service recovery, and returns predicted changes in state consistency, reachability, and safety-margin variables.

On top of the digital twin representation, DT-HERMES quantitatively evaluates the feasibility of candidate workflows. Instead of judging a workflow only by linguistic coherence, the framework scores it according to its consistency with real-time grid conditions. The workflow feasibility score is defined as

$$S_f = \lambda_1 C_s + \lambda_2 C_t + \lambda_3 C_o \tag{4}$$

where S_f denotes the overall feasibility score of a candidate workflow, C_s denotes the state consistency index measuring alignment between candidate actions and the current state estimation results, C_t denotes the topology reachability index reflecting whether the candidate workflow respects connectivity and control-path constraints, and C_o denotes the operational safety index describing the effect of the workflow

on grid operating boundaries. The coefficients λ_1, λ_2, and λ_3 are weighting factors satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The purpose of Eq. (4) is to transform the subjective notion of workflow reasonableness into a measurable criterion that can be compared across different workflow candidates. Its advantage is that it provides a common evaluation signal for both automated workflow selection and later feedback-driven revision.

Unless otherwise stated, the experiments use $\lambda_1 = 0.40$, $\lambda_2 = 0.30$, and $\lambda_3 = 0.30$. These values are selected on the validation set after expert calibration, because state consistency is slightly more important than topology reachability and operational safety in distinguishing feasible from infeasible response workflows. The candidate range $\{0.20, 0.30, 0.40, 0.50\}$ is evaluated under the constraint $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and the final setting is fixed before test-set evaluation.

In practice, the three sub-indicators in Eq. (4) may come from different layers of evidence. The state consistency index can be further expressed as

$$C_s = 1 - \frac{\|\hat{x}_t - \tilde{x}_t\|_2}{\kappa_s} \tag{5}$$

where $\hat{x}_t$ denotes the estimated operating state derived from the current monitoring and estimation system, $\tilde{x}_t$ denotes the operating state implied by the candidate workflow after twin-based simulation, $\|\cdot\|_2$ denotes the Euclidean norm, and κ_s denotes a normalization coefficient used to bound the metric. The purpose of Eq. (5) is to measure whether the intended disposal actions remain compatible with the actual estimated state of the power grid. The advantage of this definition is that it directly embeds state estimation results into workflow assessment, which is particularly important when alarm interpretation alone cannot reveal the operational consequences of a response action.

Similarly, the topology reachability index can be formulated as

$$C_t = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \chi(p) \tag{6}$$

where $\mathcal{P}$ denotes the set of required control or communication paths involved in a candidate workflow, $|\mathcal{P}|$ denotes the cardinality of that path set, and $\chi(p)$ is an indicator function that equals 1 if path p remains reachable under the current topology and security restrictions, and 0 otherwise. The purpose of Eq. (6) is to ensure that workflows are not only logically ordered but also physically and communicationally realizable. Its advantage is that it explicitly accounts for topology-dependent action feasibility, thereby reducing the risk that a generated workflow contains steps that cannot actually be executed because the required information or control path is unavailable.

The hybrid agent collaborative reasoning module is then introduced to generate and refine response workflows. Instead of assigning all responsibilities to a single large language model, DT-HERMES organizes multiple specialized agents into a coordinated reasoning structure. The local decision of the i-th agent at time t is defined as

$$u_i^{(t)} = \pi_i\left(o_i^{(t)}, m_i^{(t)}\right) \tag{7}$$

where $u_i^{(t)}$ denotes the local decision produced by the i-th agent, $\pi_i(\cdot)$ denotes its decision policy, $o_i^{(t)}$ denotes its local observation, and $m_i^{(t)}$ denotes its accessible memory context or intermediate reasoning state. The purpose of Eq. (7) is to formalize role-specialized decision making in a multi-agent environment. In DT-HERMES, semantic generation agents interpret alarm descriptions and operational guidelines to produce an initial workflow, verification agents check compliance with state-estimation consistency, topology dependencies, and safety boundaries, and coordination agents reconcile conflicts and adjust execution order. The advantage of such role specialization is that it prevents the workflow generation process from being dominated by a single reasoning perspective and improves both interpretability and robustness.

Each agent is further associated with a permission set $\mathcal{P}_i = (\mathcal{O}_i, \mathcal{B}_i, \mathcal{D}_i)$, where $\mathcal{O}_i$ defines observable state fields, $\mathcal{B}_i$ defines executable or recommendable action types, and $\mathcal{D}_i$ defines decision authority. For example, the semantic agent can propose response objectives but cannot approve control actions; the topology agent can veto unreachable communication or control paths; the safety agent can veto actions that violate operating margins; and the coordination agent can reorder or merge subtasks but cannot override a hard safety veto.

After local decisions are produced, DT-HERMES aggregates them into a coordinated global output:

$$U_t = \Psi\left(u_1^{(t)}, u_2^{(t)}, \ldots, u_N^{(t)}\right) \tag{8}$$

where U_t denotes the joint decision at time t, $\Psi(\cdot)$ denotes the multi-agent aggregation function, and N denotes the total number of participating agents. The purpose of Eq. (8) is to enforce consistency among different agent outputs so that the final workflow reflects both semantic adequacy and operational constraints. In the implemented system, $\Psi(\cdot)$ is specifically realized by the constraint-aware weighted aggregation procedure defined below, rather than by selecting among several alternative fusion mechanisms.

In the implemented DT-HERMES system, $\Psi(\cdot)$ is a constraint-aware weighted aggregation algorithm. For each candidate action a, agent M_i returns a tuple $(a, y_i(a), p_i(a), s_i(a))$, where $y_i(a) \in \{-1, 0, 1\}$ denotes reject, abstain, or support, $p_i(a) \in [0, 1]$ denotes

confidence, and $s_i(a) \in [0,1]$ denotes the agent-local safety or feasibility score. The aggregated score is

$$Q(a) = \sum_{i=1}^{N} w_i y_i(a) p_i(a) s_i(a) \tag{9}$$

where w_i is the role reliability weight. It is estimated from the validation-set action-feasibility score r_i and normalized as $w_i = r_i / \sum_{j=1}^{N} r_j$. The aggregation procedure first collects the candidate actions returned by all agents and removes duplicate candidates. It then applies a hard-veto rule: an action is discarded if any designated topology or safety agent reports a violation. Among the remaining candidates, action a is accepted only if $Q(a) > 0$ and $S_f(a) \geq \xi$. When two accepted actions conflict in resource usage, control target, or execution order, each action is associated with the priority $P(a)$ of its corresponding subtask, and the actions are ranked lexicographically by $Q(a)$ and $P(a)$. The higher-ranked action is retained, while the lower-ranked action is delayed, revised, or removed. Mutually compatible actions are retained in parallel. Therefore, Ψ is implemented as four explicit steps: candidate collection, hard-veto filtering, weighted acceptance, and priority-aware conflict elimination.

To further improve cross-agent coherence, the memory update of the i-th agent is modeled as

$$m_i^{(t+1)} = \Omega_i \left(m_i^{(t)}, o_i^{(t)}, U_t \right) \tag{10}$$

where $\Omega_i(\cdot)$ denotes the memory update function of the i-th agent. The purpose of Eq. (10) is to allow each agent to revise its internal reasoning context according to both newly observed information and the latest global coordination outcome. This design is particularly important in emergency response, because the significance of the same alarm may change after another agent identifies a topology bottleneck or a safety-boundary violation. The advantage of this formulation is that it supports iterative collaborative refinement rather than one-round independent outputs.

Once a globally coordinated workflow is obtained, DT-HERMES performs hierarchical task decomposition to convert high-level response goals into executable subtasks. Let g denote a high-level response objective. The decomposed task set is given by

$$\mathcal{T}(g) = \{\tau_1, \tau_2, \ldots, \tau_K\} \tag{11}$$

where $\mathcal{T}(g)$ denotes the set of subtasks associated with objective g, τ_k denotes the k-th subtask, and K denotes the total number of subtasks. The purpose of Eq. (11) is to represent response execution not as a monolithic procedure, but as a structured set of smaller operational units that can be assigned, checked, and adjusted separately. The advantage of such decomposition is that it improves executability and traceability, especially when a single emergency objective involves dispatching decisions, security operations, communication checks, and field actions simultaneously.

To explicitly characterize each decomposed subtask, DT-HERMES defines

$$\tau_k = (a_k, \rho_k, \delta_k, \eta_k) \tag{12}$$

where a_k denotes the concrete action of the subtask, ρ_k denotes the responsible role, δ_k denotes the prerequisite dependency relation, and η_k denotes the operational constraint or safety boundary associated with the subtask. The purpose of Eq. (12) is to ensure that decomposition outputs carry actionable semantics rather than only textual descriptions. In this formulation, a_k may correspond to alarm verification, control permission confirmation, channel switching, isolation instruction issuance, or feedback validation; ρ_k may correspond to dispatching personnel, cybersecurity operators, communication engineers, or field teams; δ_k defines which subtasks must be completed before τ_k can be executed; and η_k encodes operating constraints such as topology limits, time windows, or safety margins. The advantage of this structured representation is that it naturally supports role assignment, dependency checking, and downstream workflow execution.

Formally, the constraint term is defined as $\eta_k = (\eta_k^s, \eta_k^a, \eta_k^p, \eta_k^r)$, where η_k^s is a state constraint, η_k^a is an action constraint, η_k^p is a permission constraint, and η_k^r is a resource or time constraint. The k-th subtask is executable only if

$$\begin{aligned} \eta_k(\mathcal{Z}_t, a_k, \rho_k) = \mathbb{I}\Big[& \mathbf{h}_k(\mathcal{Z}_t, a_k) \leq \mathbf{0}, \\ & a_k \in \mathcal{B}_{\rho_k},\ \rho_k \in \mathcal{D}(a_k), \\ & t_k^{\text{start}} \geq \max_{\tau_j \in \delta_k} t_j^{\text{finish}} \Big] = 1. \end{aligned} \tag{13}$$

where $\mathbf{h}_k(\cdot) \leq \mathbf{0}$ covers voltage/current/process-state margins, communication reachability, and measurement-credibility bounds, $\mathcal{B}_{\rho_k}$ is the action set permitted for role ρ_k, and $\mathcal{D}(a_k)$ is the set of roles authorized to execute or approve action a_k.

To capture the dependency structure among subtasks more explicitly, DT-HERMES models the decomposition graph as

$$\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T) \tag{14}$$

where $\mathcal{G}_T$ denotes the task dependency graph, $\mathcal{V}_T$ denotes the set of task nodes corresponding to decomposed subtasks, and $\mathcal{E}_T$ denotes the set of directed edges representing temporal or logical precedence relations. The purpose of Eq. (14) is to transform the decomposed workflow into a graph-structured execution object that can be checked for conflicts, parallelizable branches, or cyclic inconsistencies. The advantage of such graph

modeling is that it supports both analytical verification and implementation-oriented scheduling.

To ensure that task decomposition balances response effectiveness with operational safety, DT-HERMES formulates an optimization objective over the decomposed task set:

$$\max_{\mathcal{T}(g)} J = \sum_{k=1}^{K} \omega_k v_k - \mu \sum_{k=1}^{K} c_k \tag{15}$$

where J denotes the overall objective value of the decomposition scheme, v_k denotes the contribution value of the k-th subtask to the global response objective, c_k denotes the execution or risk cost associated with that subtask, ω_k denotes the contribution weight, and μ denotes the cost suppression coefficient. The purpose of Eq. (15) is to balance response sufficiency against disturbance minimization, so that the system does not blindly favor aggressive actions that may protect cyber assets at the expense of grid stability. Here, v_k can reflect risk mitigation, critical-asset protection, or recovery effectiveness, while c_k can incorporate communication overhead, control impact scope, misoperation risk, and human intervention cost. The main advantage of this formulation is that decomposition becomes a constrained optimization process rather than a purely semantic splitting procedure.

For practical execution, the response priority of each subtask is further evaluated as

$$P_k = \alpha r_k + \beta d_k + \gamma q_k \tag{16}$$

where P_k denotes the priority score of subtask τ_k, r_k denotes the risk mitigation urgency, d_k denotes the dependency criticality, q_k denotes the operational impact severity if the subtask is delayed, and α, β, and γ are weighting coefficients satisfying $\alpha + \beta + \gamma = 1$. The purpose of Eq. (16) is to provide an ordering criterion for subtask scheduling under limited resources or time pressure. The advantage of this priority formulation is that it merges security urgency with workflow structure and operational consequence, making the execution sequence more adaptive than a simple first-generated-first-executed rule.

The default priority parameters are $\alpha = 0.45$, $\beta = 0.25$, and $\gamma = 0.30$, selected by validation-set grid search after expert calibration. The cost suppression coefficient in Eq. (15) is set to $\mu = 0.20$, which penalizes unnecessary control disturbance while preserving sufficient response coverage. The feasibility threshold in Algorithm 1 is set to $\xi = 0.75$, and the revision threshold in Eq. (19) is set to $\theta = 0.15$ after normalizing twin-state discrepancy to $[0, 1]$. These values are not tuned on the test set.

Since power grid cybersecurity incidents evolve continuously, DT-HERMES incorporates a real-time feedback-driven revision mechanism so that workflows and task structures can be updated online. Let Π_t denote the response strategy at time t. After receiving new monitoring feedback $\Delta\mathcal{X}_{t+1}$, the updated strategy is defined as

$$\Pi_{t+1} = \Gamma\left(\Pi_t, \Delta\mathcal{X}_{t+1}\right) \tag{17}$$

where Π_{t+1} denotes the revised response strategy, $\Gamma(\cdot)$ denotes the closed-loop revision operator, and $\Delta\mathcal{X}_{t+1}$ denotes the variation of the integrated emergency state from time t to time $t+1$. The purpose of Eq. (17) is to support rolling replanning in situations where alarm confidence changes, attack influence expands, local actions fail, or operating boundaries become tighter. The advantage of this mechanism is that it allows emergency disposal to remain effective even when the initial response assumptions become outdated.

To determine whether revision is necessary, DT-HERMES evaluates the discrepancy between the predicted twin evolution and the observed feedback as

$$E_{t+1} = \left\| \mathcal{Z}_{t+1}^{\text{obs}} - \mathcal{Z}_{t+1}^{\text{pred}} \right\|_2 \tag{18}$$

where E_{t+1} denotes the twin-state discrepancy at time $t+1$, $\mathcal{Z}_{t+1}^{\text{obs}}$ denotes the observed twin state constructed from real-time monitoring feedback, and $\mathcal{Z}_{t+1}^{\text{pred}}$ denotes the predicted twin state derived from the previous response strategy. The purpose of Eq. (18) is to quantify the mismatch between expected response effects and actual system evolution. The advantage of this discrepancy measure is that it gives DT-HERMES an explicit trigger for adaptive correction, rather than waiting until a workflow fails completely before intervention.

Accordingly, a revision trigger can be defined as

$$\zeta_{t+1} = \begin{cases} 1, & E_{t+1} > \theta, \\ 0, & E_{t+1} \le \theta, \end{cases} \tag{19}$$

where ζ_{t+1} denotes the revision indicator and θ denotes the discrepancy threshold for triggering replanning. The purpose of Eq. (19) is to distinguish ordinary fluctuation from meaningful deviation that requires response adjustment. Its advantage is that it prevents excessive replanning while preserving adaptability under rapidly changing attack conditions.

The discrepancy threshold θ is applied to normalized twin-state variables. For datasets without complete physical measurements, the same trigger is computed over the available state dimensions and the corresponding feature mask, so that missing physical features do not artificially reduce or enlarge E_{t+1}.

Overall, DT-HERMES is not merely a language-model-based emergency text generator, nor is it only a digital twin platform for state visualization.

Through the joint design of unified state representation, cross-layer twin mapping, hybrid agent collaboration, graph-aware hierarchical task decomposition, and feedback-driven closed-loop revision, it organizes power grid cybersecurity emergency response into a dynamic decision process that is interpretable, verifiable, and executable. The main strength of the framework lies in its ability to jointly accommodate unstructured security-oriented semantic reasoning and grid-operation-oriented engineering constraints, thereby improving both the intelligence of workflow generation and the practical feasibility of task execution in smart energy systems.

Algorithm 1 DT-HERMES for Power Grid Cybersecurity Emergency Response

Require: Alarm set $\mathcal{A}_t$, grid state $\mathcal{G}_t$, knowledge base $\mathcal{K}$, agent set $\mathcal{M} = \{M_1, \ldots, M_N\}$, thresholds ξ, θ, weights $(\lambda_1, \lambda_2, \lambda_3)$, (α, β, γ), and μ
Ensure: Emergency workflow Π_t and task set $\mathcal{T}(g)$
Initialize $\mathcal{X}_t \leftarrow (\mathcal{A}_t, \mathcal{G}_t, \emptyset)$ and $\mathcal{Z}_t \leftarrow \Phi(\mathcal{A}_t, \mathcal{G}_t)$
Extract high-level objective g from $\mathcal{A}_t$ and $\mathcal{K}$
for each agent $M_i \in \mathcal{M}$ **do**
    Generate local decision $u_i^{(t)} \leftarrow \pi_i(o_i^{(t)}, m_i^{(t)})$
end for
Aggregate decisions $U_t \leftarrow \Psi(u_1^{(t)}, u_2^{(t)}, \ldots, u_N^{(t)})$ using Eq. (9) with hard-veto checking
Compute feasibility score $S_f \leftarrow \lambda_1 C_s + \lambda_2 C_t + \lambda_3 C_o$
while $S_f < \xi$ **do**
    for each agent $M_i \in \mathcal{M}$ **do**
        Update memory $m_i^{(t)} \leftarrow \Omega_i(m_i^{(t)}, o_i^{(t)}, U_t)$
        Regenerate decision $u_i^{(t)} \leftarrow \pi_i(o_i^{(t)}, m_i^{(t)})$
    end for
    Update $U_t \leftarrow \Psi(u_1^{(t)}, u_2^{(t)}, \ldots, u_N^{(t)})$ and S_f
end while
Set workflow $\Pi_t \leftarrow U_t$ and decompose g into $\mathcal{T}(g) = \{\tau_1, \ldots, \tau_K\}$
for each subtask τ_k **do**
    Assign $(a_k, \rho_k, \delta_k, \eta_k)$ and compute priority P_k
    Check executability constraint $\eta_k(\mathcal{Z}_t, a_k, \rho_k) = 1$
end for
Optimize decomposition objective $J = \sum_{k=1}^{K} \omega_k v_k - \mu \sum_{k=1}^{K} c_k$
Build dependency graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T)$ and schedule tasks
Predict $\mathcal{Z}_{t+1}^{\text{pred}} \leftarrow F(\mathcal{Z}_t, U_t, \varepsilon_t)$ and observe $\mathcal{Z}_{t+1}^{\text{obs}}$
Compute discrepancy $E_{t+1} \leftarrow \|\mathcal{Z}_{t+1}^{\text{obs}} - \mathcal{Z}_{t+1}^{\text{pred}}\|_2$
if $E_{t+1} > \theta$ **then**
    Update strategy $\Pi_{t+1} \leftarrow \Gamma(\Pi_t, \Delta\mathcal{X}_{t+1})$
    return Π_{t+1} and updated $\mathcal{T}(g)$
end if
return Π_t and $\mathcal{T}(g)$

4. Experiments

This section evaluates DT-HERMES from three complementary perspectives. First, an overall performance evaluation is conducted to compare DT-HERMES with representative published baselines on multiple public cybersecurity and cyber-physical datasets. Second, an ablation study is performed to validate the contributions of the twin mapping module, the feasibility verification module, the hybrid agent collaboration mechanism, and the task decomposition and revision module. Third, robustness experiments are carried out under alarm missing, topology perturbation, observation delay, and compound attack conditions to assess the stability and closed-loop response capability of the proposed method in dynamic scenarios.

To provide a comprehensive evaluation, both predictive metrics and response-oriented metrics are adopted. In addition to Macro-F1 and Accuracy, this study further considers workflow feasibility, state consistency, task success rate, response recovery rate, performance degradation ratio, cumulative control risk, replanning frequency, and average response latency. These metrics are intended to reflect not only whether the method can correctly interpret cybersecurity events, but also whether it can generate executable emergency workflows and maintain reliable response performance under real-time operational constraints and dynamic perturbations.

4.1 Datasets and Experimental Settings

To cover alarm interpretation, cyber-physical state perception, and emergency workflow validation in smart energy and industrial control scenarios, six real and publicly accessible datasets are selected in this study, as summarized in Table 2. Among them, CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021 are directly downloadable public datasets, while EPIC, SWaT, and WaDi are publicly accessible academic datasets provided through official request-based access. Although EPIC, SWaT, and WaDi originate from real testbeds, the data used in this work are dataset releases rather than the platforms themselves. This distinction is important because the experiments are conducted on released benchmark data rather than on live physical test environments.

The selected datasets jointly support the experimental objectives of this paper. EPIC is the most closely related to critical energy infrastructure and is mainly used for digital twin validation and case-oriented response analysis. SWaT and WaDi provide multistage attack processes with rich sensor and actuator measurements, which are suitable for evaluating state consistency, process-aware workflow generation, and feedback-driven revision. CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021 provide diverse network and

Table 2. Public Datasets Used in the Experiments

Dataset	Scale / Duration	Feature Dimension
EPIC	4 scenarios, 96 min, 2867 records	292–391
SWaT	946,722 samples, 11 days	51
WaDi	1,221,370 samples, 16 days	123
CICIDS2017	2,830,743 flows, 5 days	78
ToN_IoT (network subset)	461,043 records	43
WUSTL-IIoT-2021	1,194,464 observations, 53 h	41

Table 3. Dataset-level task construction and response objectives

Dataset	Labels and attack categories	Sample construction	Training objective	Response objective
EPIC	Scenario ID and event label provided by the EPIC release	One synchronized cyber-physical experiment record; alarm, state, and action fields are aligned by timestamp	Joint prediction of scenario/event label y_t^a and response objective y_t^r	Verification, protection, and recovery confirmation selected from the scenario impact and response-template mapping
SWaT	Normal/attack, with the dataset attack-stage label retained when available	1 s sampled process state; contiguous attack segments remain in one data split	Binary or attack-stage recognition plus response-objective prediction	Isolation, process-state checking, and recovery confirmation according to the affected sensor/actuator group
WaDi	Normal/attack, with the dataset attack-stage label retained when available	1 s sampled process state; contiguous attack segments remain in one data split	Binary or attack-stage recognition plus response-objective prediction	Protection, measurement verification, and recovery confirmation according to the affected process group
CICIDS2017	Benign and dataset-provided network attack categories	60 s flow window with 30 s stride; the window label is derived from the attack records assigned to that window	Multi-class attack-category recognition plus response-objective prediction	Traffic filtering, service isolation, and channel or access restriction based on the attack category and affected service
ToN_IoT	Benign and dataset-provided IoT attack categories	Record-based event window; records from one attack segment are kept in the same split	Attack-type recognition plus response-objective prediction	Access restriction, protection, and affected-service verification based on the attack type
WUSTL-IIoT-2021	Benign and dataset-provided IIoT attack categories	Observation window with aligned network and service-state fields	Attack-state recognition plus response-objective prediction	Isolation, service recovery, and post-isolation verification based on the affected IIoT service

Table 4. Baseline Methods Used for Comparison

Abbreviation	Comparison Role
PS+	Explicit planning and stepwise task decomposition baseline
Reflexion	Feedback-driven language agent baseline
Self-Refine	Self-feedback iterative refinement baseline
CAMEL	Multi-agent collaboration baseline
GSCCB-SHOP2	Hierarchical task network planning baseline

Table 5. Implementation and Training Settings

Item	Setting
Operating system	Ubuntu 22.04 LTS
CPU	Intel Xeon Gold 6338, 32 cores, 2.0 GHz
GPU	NVIDIA A100 80GB
Memory	256 GB RAM
CUDA	12.1
Python version	Python 3.10
Deep learning framework	PyTorch 2.2.2
Optimizer	AdamW
Initial learning rate	2×10^{-5}
Batch size	16
Maximum epochs	10
Early stopping	Validation loss, patience = 3
Maximum sequence length	2048 tokens
Network flow window	60 s, stride 30 s
Process state sampling	1 s
Data split	7:1:2; EPIC uses 5-fold cross-validation

IIoT attack records and are used to evaluate alarm understanding, response target generation, and workflow planning under heterogeneous attack conditions. For unified modeling, all raw records are transformed into event-window-level samples and represented as triplets consisting of alarm description, system state vector, and candidate response objective.

Table 3 summarizes the labels, sample construction, training objective, and response objective for each dataset. Each event-window sample contains two supervised targets: an attack-recognition label y_t^a and a response-objective label y_t^r. The attack-recognition target preserves the native dataset label granularity used in the experiments. The response-objective target is generated deterministically from the native attack or scenario label, the affected feature or asset group,

and the domain response-template library; it is not manually assigned after model prediction.

For data partitioning, CICIDS2017, ToN_IoT, WUSTL-IIoT-2021, SWaT, and WaDi are divided into training, validation, and test sets with a ratio of 7:1:2 using attack-aware stratified sampling. EPIC is evaluated with five-fold cross-validation due to its relatively small size but strong scenario relevance. For time-series process data, continuous attack segments are kept within the same split to reduce temporal leakage. Numerical features are normalized by z-score standardization, while categorical fields are converted into structured event descriptions through label encoding and semantic templates.

Because the six datasets do not contain identical feature sets, the input alignment is performed through a shared schema rather than by forcing all datasets into the same raw feature dimension. The shared schema contains four blocks: alarm semantics, cyber/network statistics, process or grid-state variables, and availability/credibility masks. If a dataset lacks a block, the corresponding feature values are masked by $M_t = 0$ and excluded from distance, consistency, and loss calculations. For example, CICIDS2017 has rich network-flow variables but lacks physical power-system measurements, so its state consistency is computed over cyber reachability and service-state features rather than voltage or power-flow variables. This masking strategy prevents absent features from being confused with observed zero values.

Five published baseline methods are selected for comparison, as listed in Table 4, namely PS+ [27], Reflexion [28], Self-Refine [29], CAMEL [10], and GSCCB-SHOP2 [30]. All of them are formally published methods rather than preprint-only approaches. In the experiments, the LLM-based baselines receive the same event input, external knowledge source, and tool interface as DT-HERMES to ensure a fair comparison. For GSCCB-SHOP2, the same incident-state input is converted into a hierarchical task planning representation under equivalent operational constraints.

For training fairness, only the lightweight task heads, mapping adapters, scoring modules, and role-reliability weights are trained. The base LLM is not fine-tuned; it is used through the same prompting interface for DT-HERMES, Reflexion, Self-Refine, and CAMEL. The digital-twin module is not trained end-to-end from test data; its state transition rules and normalization parameters are calibrated on the training and validation splits only. Baseline methods use the same training, validation, and test partitions, the same external knowledge base, the same maximum sequence length, and the same stopping criterion. For PS+, Reflexion, Self-Refine, and CAMEL, the response-objective prediction head is trained with the same AdamW protocol as DT-HERMES when a trainable head is required; for GSCCB-SHOP2, symbolic operators are generated from the same response-template library and validated under the same constraints.

The implementation settings are kept consistent across methods as much as possible, as summarized in Table 5. For modules requiring parameter optimization, the AdamW optimizer is adopted with an initial learning rate of 2×10^{-5}, a batch size of 16, and a maximum of 10 training epochs. Early stopping is applied based on validation loss. Network-flow datasets are processed with a 60 s sliding window and a 30 s step size, while process-control datasets use a 1 s state sampling interval. In addition to offline prediction performance, online response cost is also recorded, including average inference latency, revision frequency, and token consumption per workflow generation round.

Accuracy and Macro-F1 are reported for the attack-recognition target y_t^a unless explicitly stated otherwise. Response-objective generation is evaluated separately by Workflow Feasibility, State Consistency, Task Success Rate, Revision Effectiveness, Response Recovery Rate, Performance Degradation Ratio, Cumulative Control Risk, Replanning Frequency, and Average Response Latency. Workflow Feasibility measures the proportion of generated workflows whose $S_f \geq \xi$ and whose subtasks satisfy Eq. (13). State Consistency measures agreement between predicted and observed twin states over available feature dimensions. Task Success Rate measures whether the generated response objective and all mandatory subtasks are completed under dependency, permission, and safety constraints.

4.2 Overall Performance Evaluation

To evaluate the overall effectiveness of DT-HERMES, this subsection analyzes the results from two perspectives. First, the Macro-F1 trends under different training data ratios are compared across six datasets to assess data efficiency and generalization stability. Second, under the full training data setting, the methods are further compared in terms of Accuracy, Workflow Feasibility, State Consistency, and Task Success Rate, so as to examine their overall capability in event understanding, workflow generation, and task execution.

As shown in Fig. 2, the Macro-F1 scores of all methods increase as the training data ratio grows, indicating that additional training samples generally improve attack understanding and response modeling performance. However, DT-HERMES consistently achieves the best overall trend across the six datasets and maintains stronger stability in low-data regimes. This advantage is particularly clear on EPIC, WaDi, and WUSTL-IIoT-2021, where the proposed method obtains

Table 6. Overall comparison on six datasets under full training data ratio

Dataset	Method	Accuracy (%)	Workflow Feasibility (%)	State Consistency (%)	Task Success Rate (%)
EPIC	PS+	76.84	71.32	69.48	67.25
	Reflexion	78.63	74.85	72.91	70.88
	Self-Refine	79.12	75.41	72.36	71.44
	CAMEL	81.37	79.68	77.94	75.83
	GSCCB-SHOP2	73.95	74.96	75.28	72.17
	DT-HERMES	**84.56**	**85.92**	**84.37**	**82.41**
SWaT	PS+	87.42	80.73	79.68	77.95
	Reflexion	89.18	84.11	82.45	80.86
	Self-Refine	89.56	84.68	81.97	81.24
	CAMEL	91.24	88.36	86.91	85.44
	GSCCB-SHOP2	84.37	81.95	82.84	79.26
	DT-HERMES	**93.08**	**92.47**	**91.13**	**89.86**
WaDi	PS+	82.16	75.48	73.86	71.74
	Reflexion	84.03	78.52	76.91	74.68
	Self-Refine	84.47	79.05	76.34	75.13
	CAMEL	86.65	83.24	81.36	79.84
	GSCCB-SHOP2	79.28	76.89	77.51	74.93
	DT-HERMES	**89.21**	**88.34**	**86.97**	**84.92**
CICIDS2017	PS+	89.83	79.64	78.12	76.95
	Reflexion	91.26	82.53	80.71	79.42
	Self-Refine	91.48	83.17	80.35	79.96
	CAMEL	93.07	86.54	84.28	83.46
	GSCCB-SHOP2	87.24	77.85	78.63	76.38
	DT-HERMES	**95.11**	**91.43**	**89.62**	**88.17**
ToN_IoT	PS+	86.38	77.21	75.86	74.23
	Reflexion	88.11	80.34	78.62	77.05
	Self-Refine	88.46	80.92	78.11	77.42
	CAMEL	90.35	84.58	82.51	81.44
	GSCCB-SHOP2	83.56	76.43	77.08	74.85
	DT-HERMES	**92.74**	**89.37**	**87.93**	**86.12**
WUSTL-IIoT-2021	PS+	84.57	73.88	72.71	69.91
	Reflexion	86.44	77.47	75.84	73.28
	Self-Refine	86.73	78.09	75.25	73.94
	CAMEL	88.19	82.14	79.72	78.07
	GSCCB-SHOP2	80.92	74.85	75.64	72.19
	DT-HERMES	**91.06**	**87.89**	**85.53**	**83.51**

earlier and larger performance gains than the other baselines. Such a result suggests that DT-HERMES does not rely only on data scale itself, but can further exploit system-state constraints and prior response knowledge through digital twin guidance and hybrid agent collaboration. By contrast, PS+, Reflexion, and Self-Refine also improve as more data become available, but their gains tend to saturate earlier. CAMEL performs better than the other baselines in most cases, confirming the usefulness of multi-agent collaboration for complex response tasks, but it still remains below DT-HERMES on several datasets due to the lack of explicit cyber-physical state alignment.

Table 6 further reports the quantitative comparison under the full training data setting. DT-HERMES achieves the best results on all six datasets across the four reported metrics. Although its improvement in Accuracy is already stable and consistent, the advantage becomes more evident in Workflow Feasibility, State Consistency, and Task Success Rate, which are more directly related to the target problem of this study. On EPIC, SWaT, and WaDi, the gains of DT-HERMES in State Consistency and Task Success Rate are especially notable, showing that its improvement is not limited to better event recognition, but more importantly lies in generating workflows that better satisfy operational constraints and decomposing them into executable subtasks. This observation is consistent with the design of the digital twin validation, topology-aware verification, and hierarchical task decomposition modules.

A similar pattern can also be observed on CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021. In these network and IIoT scenarios, the proposed method still outperforms the baselines, while its larger margin mainly appears in Workflow Feasibility and State Consistency rather than only in conventional recognition accuracy. This indicates that when attack detection performance is already relatively strong, the real challenge of emergency response lies in converting

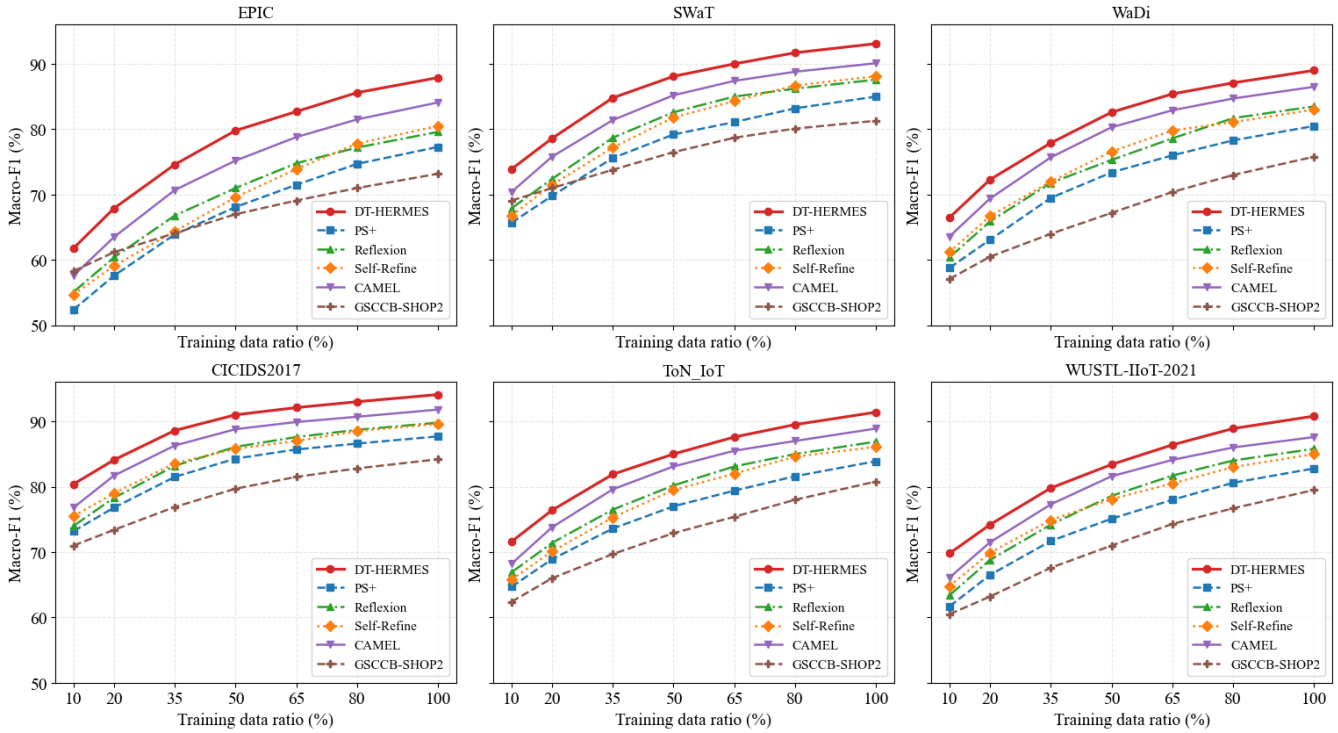

Figure 2. Macro-F1 comparison of DT-HERMES and baseline methods under different training data ratios on six public datasets

detected cyber incidents into executable workflows consistent with operating states and control constraints. Overall, the results in Fig. 2 and Table 6 demonstrate that DT-HERMES provides a more balanced and effective s olution f or i ntegrating e vent perception, state estimation constraints, and emergency task organization in smart energy systems.

4.3 Ablation Study and Mechanism Validation

To further verify the effectiveness o f t he c ore components in DT-HERMES, an ablation study is conducted on six datasets. According to the method design presented in the previous section, the complete framework consists of the twin mapping module (TM), the feasibility verification m odule (FV), t he h ybrid agent collaboration module (HA), and the task decomposition and revision module (TDR). Based on the full model, four degraded variants, namely *w/o TM*, *w/o FV*, *w/o HA*, and *w/o TDR*, are constructed and compared with the complete DT-HERMES, so as to analyze the contribution of each module to emergency workflow modeling and task decomposition.

As shown in Fig. 3, the complete DT-HERMES achieves the best results on all six datasets in terms of Workflow Feasibility and Task Success Rate, indicating that the coordinated integration of all modules substantially improves both the executability of the generated workflows a nd t he s uccess o f decomposed task execution. In particular, removing TM leads to a clear reduction in Workflow Feasibility on EPIC, SWaT, and WaDi, which suggests that the digital-twin-oriented mapping mechanism is especially important for scenarios with stronger coupling between cyber events and physical operating states. By comparison, the impact of removing TM is relatively smaller on CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021, although a consistent performance drop can still be observed, indicating that state alignment remains beneficial even in more network-oriented attack scenarios.

It can also be observed from Fig. 3 that removing FV causes the most obvious degradation in Workflow Feasibility across the six datasets. This observation is consistent with the role of FV in the proposed framework, namely filtering out response workflows that violate topology constraints, state boundaries, or operational safety requirements. In contrast, removing TDR results in the largest decrease in Task Success Rate on all datasets, especially in EPIC, SWaT, and WaDi, where multi-stage execution and iterative update are more critical. This indicates that hierarchical task decomposition and revision are essential for transforming high-level response workflows into practically executable subtasks.

Table 7 further reports the ablation results in terms of State Consistency, Revision Effectiveness, and Average Response Latency. Removing TM leads to a clear decline

Table 7. Ablation study of DT-HERMES on six datasets with additional metrics

Dataset	Variant	State Consistency (%)	Revision Effectiveness (%)	Avg. Response Latency (s)
EPIC	w/o TM	75.3	71.6	1.72
	w/o FV	77.1	73.4	1.69
	w/o HA	79.6	75.2	1.84
	w/o TDR	81.4	63.8	1.51
	DT-HERMES	**84.4**	**82.7**	2.06
SWaT	w/o TM	82.6	80.1	1.58
	w/o FV	84.3	81.7	1.55
	w/o HA	86.5	83.9	1.71
	w/o TDR	87.8	70.6	1.42
	DT-HERMES	**91.2**	**89.4**	1.93
WaDi	w/o TM	78.5	75.4	1.66
	w/o FV	80.2	76.9	1.63
	w/o HA	82.1	78.3	1.79
	w/o TDR	83.6	66.8	1.48
	DT-HERMES	**87.0**	**84.8**	2.01
CICIDS2017	w/o TM	84.8	80.7	1.21
	w/o FV	86.1	82.4	1.18
	w/o HA	86.9	83.2	1.34
	w/o TDR	88.1	71.5	1.09
	DT-HERMES	**89.6**	**88.1**	1.55
ToN_IoT	w/o TM	82.5	78.6	1.37
	w/o FV	83.8	79.9	1.34
	w/o HA	82.9	80.8	1.49
	w/o TDR	86.1	68.7	1.16
	DT-HERMES	**87.9**	**86.3**	1.71
WUSTL-IIoT-2021	w/o TM	80.7	76.8	1.33
	w/o FV	81.9	78.1	1.31
	w/o HA	81.4	79.2	1.46
	w/o TDR	84.3	67.1	1.12
	DT-HERMES	**85.5**	**83.9**	1.68

in State Consistency, particularly on EPIC and SWaT, showing that the twin mapping module plays a primary role in aligning cyber incident semantics with real-time operating states. Removing FV also reduces State Consistency, although its values are generally slightly higher than those of *w/o TM*, suggesting that feasibility verification mainly performs constraint-aware filtering after workflow generation rather than directly handling cross-layer state mapping. By comparison, removing HA causes moderate degradation in both State Consistency and Revision Effectiveness, indicating that hybrid multi-agent collaboration contributes to more coherent reasoning and better integration of intermediate feedback, especially in datasets with more heterogeneous attack semantics such as ToN_IoT and WUSTL-IIoT-2021.

Another notable observation from Table 7 is that *w/o TDR* consistently produces the lowest Revision Effectiveness, while at the same time requiring the lowest Average Response Latency. This is reasonable because without task decomposition and feedback-driven revision, the system no longer performs iterative restructuring or execution-level correction, which reduces computational cost but significantly weakens adaptability under state changes and alarm updates. In contrast, the complete DT-HERMES does not achieve the lowest latency, yet it consistently provides the highest State Consistency and Revision Effectiveness, indicating that the additional computational overhead is justified by substantial gains in execution quality and closed-loop adaptability.

Overall, the results in Fig. 3 and Table 7 show that the different components of DT-HERMES play distinct but complementary roles. TM mainly improves cross-layer state alignment, FV guarantees workflow feasibility, HA enhances collaborative reasoning under complex semantic conditions, and TDR directly supports executable task organization and closed-loop revision. The full model consistently achieves the best or the most stable performance across all six datasets, demonstrating that these components jointly contribute to the closed-loop emergency response capability required for cybersecurity incident handling in smart energy systems.

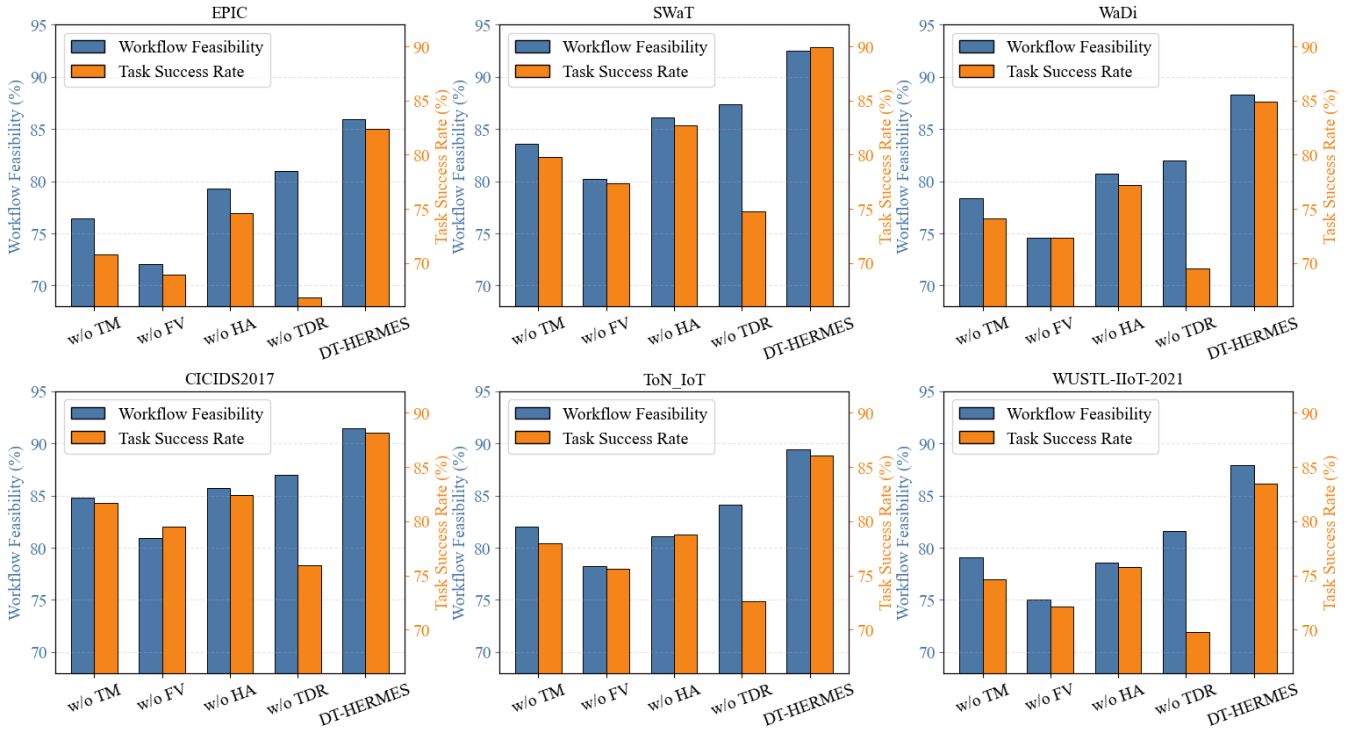

Figure 3. Ablation comparison of DT-HERMES and its variants on six datasets in terms of workflow feasibility and task success rate

4.4 Parameter Sensitivity Analysis

To examine whether the reported conclusions depend on a narrow parameter setting, a sensitivity analysis is conducted for the main weighting coefficients and decision thresholds. The tested parameters include the feasibility weights $(\lambda_1, \lambda_2, \lambda_3)$, the task-priority weights (α, β, γ), the decomposition cost coefficient μ, the feasibility threshold ξ, and the revision threshold θ. For each test, one parameter group is varied while the remaining settings are fixed at the default values reported in Section 3. The analysis is conducted on the validation and test splits of EPIC, SWaT, WaDi, CICIDS2017, ToN_IoT, and WUSTL-IIoT-2021, and the averaged test results are reported in Table 8.

The results show that moderate variations in the weighting coefficients and decision thresholds do not reverse the main conclusions. DT-HERMES remains consistently stronger than the best baseline in Table 6 under all tested settings. Increasing ξ improves workflow feasibility slightly but may reduce task success rate because some useful but marginal workflows are rejected. Lowering θ triggers more frequent revision and slightly improves state consistency, but the gain is limited because excessive replanning also increases coordination overhead. These observations indicate that the reported advantage of DT-HERMES is mainly produced by the joint design of twin mapping, feasibility verification, agent aggregation, and task decomposition, rather than by a single carefully tuned coefficient.

4.5 Robustness and Closed-Loop Response Analysis

To further evaluate the adaptability of DT-HERMES under dynamic disturbance conditions, this subsection compares different methods under four representative perturbation settings, namely alarm missing, topology perturbation, observation delay, and compound attack. Different from the previous subsections, which mainly focus on overall performance and component contribution under relatively static settings, this subsection emphasizes robustness and recovery capability when monitoring information is incomplete, state estimation is disturbed, and control conditions vary over time. Figure 4 presents the heatmaps of Response Recovery Rate on six datasets under the four perturbation types, while Tables 9 and 10 further report the corresponding Performance Degradation Ratio, Cumulative Control Risk, and Replanning Frequency on EPIC and SWaT, respectively.

As shown in Fig. 4, the recovery capability of all methods decreases as the disturbance becomes more challenging, especially when moving from alarm missing to compound attack. Nevertheless, DT-HERMES consistently achieves the highest or most stable Response Recovery Rate across the six datasets. This advantage is particularly evident on EPIC, SWaT,

Table 8. Sensitivity analysis of DT-HERMES under different parameter settings

Parameter setting	Accuracy (%)	Workflow Feasibility (%)	State Consistency (%)	Task Success Rate (%)
Default setting	90.96	89.24	87.59	85.83
$\lambda_1 = 0.30, \lambda_2 = 0.40, \lambda_3 = 0.30$	90.68	88.73	86.84	85.21
$\lambda_1 = 0.30, \lambda_2 = 0.30, \lambda_3 = 0.40$	90.71	88.91	87.06	85.34
$\alpha = 0.35, \beta = 0.35, \gamma = 0.30$	90.83	88.96	87.31	85.46
$\alpha = 0.35, \beta = 0.25, \gamma = 0.40$	90.79	88.82	87.28	85.39
$\mu = 0.10$	90.91	88.51	86.97	85.18
$\mu = 0.30$	90.86	88.87	87.43	85.37
$\xi = 0.70$	90.92	88.34	86.88	85.02
$\xi = 0.80$	90.74	89.02	87.51	85.11
$\theta = 0.10$	90.89	89.01	87.73	85.62
$\theta = 0.20$	90.81	88.47	86.95	85.09

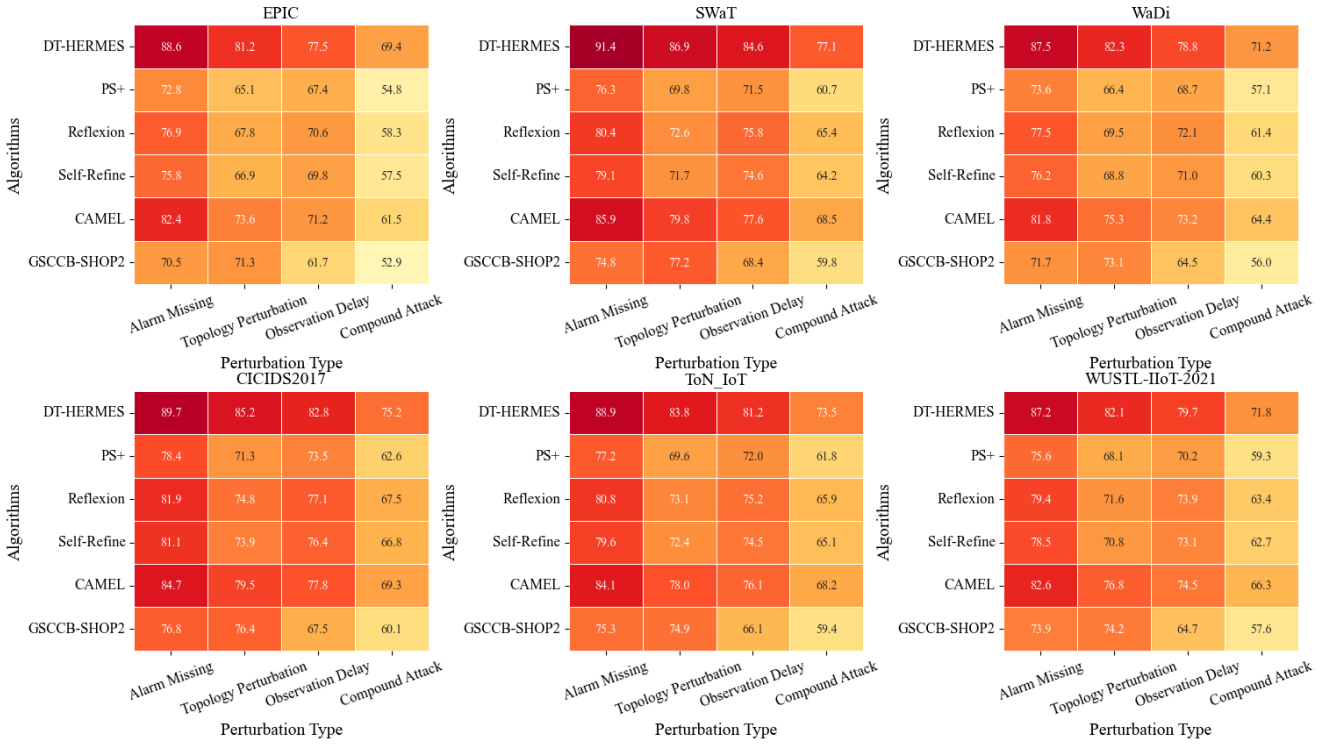

Figure 4. Heatmap comparison of response recovery rate under different perturbation conditions across six datasets and six methods

and WaDi, where cyber events are more strongly coupled with physical operating states and process dynamics. Such results indicate that the digital-twin-guided feedback mechanism enables DT-HERMES to maintain more effective workflow reconstruction even when alarms are incomplete or system states deviate from their expected trajectories. Although CAMEL generally performs better than PS+, Reflexion, Self-Refine, and GSCCB-SHOP2, reflecting the benefit of multi-agent collaboration in complex response tasks, it still remains consistently below DT-HERMES in the most challenging perturbation scenarios, especially under compound attack conditions.

Tables 9 and 10 further support this observation. On both EPIC and SWaT, DT-HERMES achieves the lowest Performance Degradation Ratio under all four perturbation types, indicating that its overall response quality deteriorates the least after disturbances are introduced. At the same time, DT-HERMES also yields the lowest Cumulative Control Risk, which suggests that its performance gain is not obtained through more aggressive control behavior, but through safer and more effective workflow revision. This property is particularly important under topology perturbation and compound attack conditions, where improper or overly aggressive control actions may introduce additional operational risk.

Another notable trend in Tables 9 and 10 is that DT-HERMES usually has a higher Replanning Frequency than the baseline methods, whereas GSCCB-SHOP2 tends to have the lowest one. This behavior is consistent with the mechanisms of the methods. DT-HERMES

Table 9. Robustness comparison under different perturbation conditions on EPIC

Perturbation	Method	PDR (%)↓	CCR ↓	RF ↑
Alarm Missing	PS+	18.7	0.46	1.2
	Reflexion	15.9	0.41	1.4
	Self-Refine	16.6	0.42	1.3
	CAMEL	12.4	0.35	1.6
	GSCCB-SHOP2	20.8	0.39	0.8
	DT-HERMES	**9.6**	**0.28**	**1.9**
Topology Perturbation	PS+	24.9	0.58	1.1
	Reflexion	21.7	0.52	1.3
	Self-Refine	22.4	0.53	1.2
	CAMEL	17.5	0.43	1.5
	GSCCB-SHOP2	19.8	0.40	0.9
	DT-HERMES	**13.8**	**0.31**	**1.8**
Observation Delay	PS+	21.3	0.50	1.2
	Reflexion	18.6	0.45	1.4
	Self-Refine	19.4	0.46	1.4
	CAMEL	16.1	0.39	1.6
	GSCCB-SHOP2	25.7	0.48	0.7
	DT-HERMES	**14.5**	**0.33**	**1.9**
Compound Attack	PS+	31.8	0.71	1.4
	Reflexion	28.1	0.66	1.7
	Self-Refine	28.9	0.67	1.6
	CAMEL	23.6	0.57	2.0
	GSCCB-SHOP2	34.2	0.63	0.9
	DT-HERMES	**19.4**	**0.44**	**2.3**

Table 10. Robustness comparison under different perturbation conditions on SWaT

Perturbation	Method	PDR (%)↓	CCR ↓	RF ↑
Alarm Missing	PS+	15.2	0.39	1.1
	Reflexion	12.8	0.35	1.3
	Self-Refine	13.5	0.36	1.2
	CAMEL	10.1	0.30	1.5
	GSCCB-SHOP2	17.4	0.33	0.8
	DT-HERMES	**7.9**	**0.24**	**1.8**
Topology Perturbation	PS+	20.6	0.49	1.0
	Reflexion	17.8	0.45	1.2
	Self-Refine	18.5	0.46	1.2
	CAMEL	14.3	0.37	1.4
	GSCCB-SHOP2	16.1	0.35	0.9
	DT-HERMES	**11.2**	**0.28**	**1.7**
Observation Delay	PS+	17.9	0.43	1.1
	Reflexion	15.3	0.39	1.3
	Self-Refine	16.1	0.40	1.3
	CAMEL	12.8	0.34	1.5
	GSCCB-SHOP2	21.5	0.41	0.7
	DT-HERMES	**10.6**	**0.27**	**1.8**
Compound Attack	PS+	27.4	0.61	1.3
	Reflexion	23.9	0.56	1.6
	Self-Refine	24.6	0.57	1.5
	CAMEL	19.8	0.48	1.9
	GSCCB-SHOP2	29.1	0.54	0.9
	DT-HERMES	**16.2**	**0.38**	**2.1**

actively triggers workflow restructuring according to digital twin feedback and changing operating states, which leads to more frequent replanning but also higher recovery rates and lower cumulative control risk. By contrast, GSCCB-SHOP2 shows relatively conservative planning behavior and therefore lower replanning frequency, yet its limited semantic flexibility and weak feedback-driven revision ability result in insufficient recovery capability under observation delay and compound attack conditions. A similar limitation can also be observed for Reflexion and Self-Refine. Although these methods can perform certain forms of feedback-based refinement, their adjustment mainly remains at the language or strategy level, and cannot fully align the revised workflow with physical operating states and execution constraints in the way DT-HERMES does.

Overall, the results in Fig. 4 and Tables 9–10 show that DT-HERMES exhibits stronger consistency and stability under different d isturbance c onditions. Its advantage is reflected not only in higher recovery rates, but also in lower performance degradation and lower cumulative control risk. These findings demonstrate that the proposed method can more effectively exploit real-time monitoring and state feedback to support closed-loop emergency response under coupled cyber-physical variations, making it better suited to smart energy systems oriented toward real-time monitoring, state estimation, and control.

5. Conclusion

This paper proposed DT-HERMES, a digital-twin-guided emergency workflow modeling and task decomposition method for power grid cybersecurity response based on LLM-MAS hybrid agents. By integrating cyber alarms, operating states, and emergency actions into a unified f ramework, t he p roposed m ethod supports more executable and adaptive closed-loop response under operational constraints.Experimental results on six public datasets showed that DT-HERMES consistently achieved superior performance in workflow feasibility, state consistency, task success rate, and robustness under multiple disturbance conditions. These findings demonstrate the effectiveness o f c ombining digital twin guidance, hybrid agent collaboration, and hierarchical task decomposition for smart energy system cybersecurity response.Future work will focus on larger-scale deployment and more efficient human-in-the-loop emergency coordination.

Acknowledgment

This work was supported by State Grid Corporation of China Science and Technology Project Funding (520940250020).

References

[1] Yang T, Liu Y, Li W. Attack and defence methods in cyber-physical power system. IET Energy Systems Integration. 2022;4(2):159-70.

[2] Alsuwian T, Shahid Butt A, Amin AA. Smart grid cyber security enhancement: Challenges and solutions—A review. Sustainability. 2022;14(21):14226.

[3] Zhou Z, Zhang X, Zhang J, Taylor G. Comprehensive review on dynamic state estimation techniques with cybersecurity applications. IET Smart Grid. 2024;7(4):370-85.

[4] Mo Y, Kim THJ, Brancik K, Dickinson D, Lee H, Perrig A, et al. Cyber–physical security of a smart grid infrastructure. Proceedings of the IEEE. 2011;100(1):195-209.

[5] Nguyen S, Abdelhakim M, Kerestes R. Survey paper of digital twins and their integration into electric power systems. In: 2021 IEEE power & energy society general meeting (PESGM). IEEE; 2021. p. 01-5.
[6] Arjmand M, Akbari B, Ghaeini HR. From voltage to vulnerability: A comprehensive survey of dynamic security risk assessment techniques in smart grids: M. Arjmand et al. International Journal of Information Security. 2026;25(1):7.
[7] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems. 2022;35:24824-37.
[8] Shafiulla S, Jena MK. A dynamic state estimation-based cyberattack detection scheme to supervise legacy pilot protection operation. IEEE Transactions on Information Forensics and Security. 2025;20:3677-88.
[9] Swetha T, Kumaran U, Meena V, Hameed IA. Leveraging AI for enhanced cybersecurity: a comprehensive review. Discover Applied Sciences. 2025;7(6):584.
[10] Li G, Hammoud H, Itani H, Khizbullin D, Ghanem B. Camel: Communicative agents for" mind" exploration of large language model society. Advances in neural information processing systems. 2023;36:51991-2008.
[11] Mansour DEA, Numair M, Zalhaf AS, Ramadan R, Darwish MM, Huang Q, et al. Applications of IoT and digital twin in electrical power systems: A comprehensive survey. IET Generation, Transmission & Distribution. 2023;17(20):4457-79.
[12] Salim DJN. Modeling and Simulating Cyber Attacks Using Attack Trees and Security Testing Tools: A Case Study of an ICT Department. Jurnal Ilmiah Multidisiplin Indonesia (JIM-ID). 2026;5(03):698-708.
[13] Amin T, Sarkar S, Song G, Yang M, Pasman H. Cyber-physical energy system integrated security & safety. In: Methods in Chemical Process Safety. vol. 9. Elsevier; 2025. p. 341-70.
[14] Gnanasekaran V, Fatima U, Glas M, Heegaard PE. A Model-Based Framework for Developing Security-Safety Incident Response Plans: V. Gnanasekaran et al. International Journal of Information Security. 2025;24(6):229.
[15] Mo Q, Wei S, Zuo Y, Jiang C, Dai F, Liu C. Construction of Conflict-Free and Efficient Cross-Organization Emergency Response Processes: A Petri Net-Based Approach. IEEE/CAA Journal of Automatica Sinica. 2026;13(6):1492-511.
[16] Wang M, Li K, Lu Y, Zhang F, Ma J, Qiao Y. Time-aware cybersecurity knowledge graph reasoning method for vulnerability analysis. IEEE Transactions on Automation Science and Engineering. 2025;23:3547-58.
[17] Guerra PAC, Nunes RC, de Lima Silva LA. A cases and clusters framework for recording, retrieving, and reusing response plans in structured cybersecurity incident management. Engineering Applications of Artificial Intelligence. 2025;160:111831.
[18] Jung JW, Lee SW. Security requirement recommendation method using case-based reasoning to prevent advanced persistent threats. Applied Sciences. 2023;13(3):1505.
[19] Kabir MR, Halder D, Ray S. Digital twins for IoT-driven energy systems: A survey. IEEE Access. 2024;12:177123-43.
[20] Kandasamy NK, Venugopalan S, Wong TK, Leu NJ. An electric power digital twin for cyber security testing, research and education. Computers and Electrical Engineering. 2022;101:108061.
[21] He Y, Wang J, Yang C, Shi D. A graph and diffusion theory-based approach for localization and recovery of false data injection attacks in power systems. Electric Power Systems Research. 2025;239:111184.
[22] Yan Y, Kunhui Y. Novel cyber-physical architecture for optimal operation of renewable-based smart city considering false data injection attacks: Digital twin technologies for smart city infrastructure management. Sustainable Energy Technologies and Assessments. 2024;65:103733.
[23] Nafees MN, Saxena N, Cardenas A, Grijalva S, Burnap P. Smart grid cyber-physical situational awareness of complex operational technology attacks: A review. ACM computing surveys. 2023;55(10):1-36.
[24] Binyamin SS, Ben Slama S. Multi-agent systems for resource allocation and scheduling in a smart grid. Sensors. 2022;22(21):8099.
[25] Zou Y. Research on the Construction and Optimization Algorithm of Cybersecurity Knowledge Graphs Combining Open Information Extraction with Graph Convolutional Networks. In: 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS). IEEE; 2025. p. 1-5.
[26] Kampourakis KE, Gkioulos V, Kavallieratos G, Lin JC. Digital Twin-Enabled Incident Detection and Response: A Systematic Review of Critical Infrastructures Applications: KE Kampourakis et al. International Journal of Information Security. 2025;24(5):194.
[27] Wang L, Xu W, Lan Y, Hu Z, Lan Y, Lee RKW, et al. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In: Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers); 2023. p. 2609-34.
[28] Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao S. Reflexion: Language agents with verbal reinforcement learning. Advances in neural information processing systems. 2023;36:8634-52.
[29] Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, et al. Self-refine: Iterative refinement with self-feedback. Advances in neural information processing systems. 2023;36:46534-94.
[30] Qi C, Wang D, Muñoz-Avila H, Zhao P, Wang H. Hierarchical task network planning with resources and temporal constraints. Knowledge-Based Systems. 2017;133:17-32.