Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.088952

ARTICLE

Disturbed Dynamic Analysis and Robust Decision-Making Control of Complex Networks

Xiusen Wang[1,*], **Zheng Fang**[2] **and Jie Chen**[2,*]

[1]The School of Mathematics and Statistics, Ningbo University, Ningbo, China
[2]The College of Electronic Engineering, National University of Defense Technology, Hefei, China

*Corresponding Authors: Xiusen Wang. Email: wxs04010129@163.com; Jie Chen. Email: jchen202209@163.com

Received: 12 July 2026; Accepted: 18 August 2026; Published: 15 September 2026

ABSTRACT: Complex networks in cyber–physical, transportation, and information infrastructures operate under topology variations, unmeasured disturbances, and limited actuation. This paper proposes robust disturbance-aware data-driven decision control (R-D3C), which couples a sliding-window graph-regularized estimator, disturbance-envelope adaptation, sparse intervention allocation, receding-horizon optimization, and a robust safety projection. The theory directly bounds the dynamic prediction regret of the implemented sliding-window estimator. A checkable sufficient condition for safety-filter feasibility is coupled with an explicit slack-and-backup fallback when the strict projection is infeasible. Practical input-to-state stability and sparse-allocation risk reduction are established. The nominal comparison uses 30 paired runs with standard deviations and significance tests, while the separate deployment-impairment benchmark uses 20 runs; evaluation also includes the IEEE 39-bus topology and timing tests up to 1000 nodes. R-D3C achieves the lowest constraint-violation rate; its residual error is statistically comparable with adaptive MPC while preserving an explicit safety layer.

KEYWORDS: Complex networks; network dynamics; robust network control; data-driven modeling; optimization in networked systems

1 Introduction

Complex networks arise in power grids, transportation systems, industrial Internet-of-Things platforms, communication infrastructures, and cooperative robots. Their control is difficult because interactions are sparse and heterogeneous, topology information can change, disturbances are only partially observed, and the available intervention budget is usually much smaller than the network size. Small-world structure remains a useful reproducible model for local interaction and long-range shortcuts [1], while data-driven network control and structural controllability provide principled tools for selecting informative actuation channels [2,3].

Robust data-driven feedback design can convert noisy trajectories into certified controllers [4], and recent learning-based control studies exploit graph structure or side information to improve adaptation [5,6]. Complementary controllability-aware studies address sparse actuation and structural control configuration in large-scale networks [7–10], but they do not couple online disturbance estimation, dynamic actuator reallocation, and a robust safety projection. Broader network-decision research studies edge/cloud architectures and offloading [11–13], game-theoretic resource allocation [14,15], and learning-based scheduling [16,17]. Two CMC studies are retained for directly related perspectives on multi-agent formation control and AI-enabled

traffic management in networked systems [18,19]. These works motivate online network decisions, but their objectives are latency, energy, or network-level performance rather than closed-loop state safety. Control-oriented surveys are closer to the present problem [20]; surveys on task-oriented mobile networks clarify the deployment context [21], but do not provide disturbance-certified actuator allocation. However, three gaps remain. First, a learned predictor and the estimator analyzed in the theory must be identical. Second, safety-filter feasibility cannot be left as an unqualified assumption, especially after abrupt topology changes. Third, sparse network-control studies often report only synthetic, single-run averages without statistical variability, real benchmark topology, or large-scale timing.

Recent resilient control provides useful complementary perspectives. Robust distributed MPC explicitly handles disturbances and denial-of-service interruptions [22], whereas resilient fuzzy adaptive control addresses input constraints, nonlinear uncertainty, and false-data injection in multi-agent systems [23]. These methods focus on fixed agent-level control protocols or distributed prediction under specified attack models. R-D3C instead addresses the joint online problem of identifying changing network dynamics, estimating an uncertainty envelope, reallocating a limited actuator set, and certifying the final action. The distinction is important: robust MPC alone does not decide which nodes should be actuated, and a graph policy alone does not certify the applied input.

The proposed R-D3C framework uses a graph-regularized sliding window to estimate local dynamics and residual energy. A disturbance-sensitive score then combines state risk, local disagreement, topology, and recent residuals. A greedy budgeted allocator chooses the active nodes, a receding-horizon policy generates a nominal command, and a robust projection verifies the Lyapunov decrease. If the strict constraint is infeasible, a penalized slack problem and a saturated backup action are invoked; the paper clearly distinguishes strict certification from best-effort fallback operation.

The contributions are summarized as follows. (1) We formulate a unified identification–allocation–control problem in which the actuator set changes with the measured disturbance path. (2) We derive practical stability and sparse risk-reduction guarantees and establish a dynamic-regret bound for the exact sliding-window ridge estimator used in the algorithm and experiments. (3) We provide a verifiable safety-feasibility condition, radius inflation after topology changes, and an explicit fallback. (4) We report reproducible baseline settings, a 30-paired-run nominal comparison with significance tests, a separate 20-run deployment-impairment benchmark, an IEEE 39-bus topology experiment [24], and runtime scaling to 1000 nodes.

The remainder of this paper is organized as follows. Section 2 defines the disturbed network problem. Section 3 develops identification and robust-set construction. Section 4 presents R-D3C. Section 5 gives the theoretical analysis. Section 6 reports the experiments, and Section 7 concludes the paper.

Table 1 summarizes the principal notation used throughout the manuscript.

Table 1: Notation and definitions.

Symbol	Definition
$\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$	Directed or undirected network at time k with $n = \lvert\mathcal{V}\rvert$ nodes.
A_k	Weighted adjacency matrix, possibly time-varying.
L_k	Graph Laplacian associated with A_k.
x_k	Stacked network state in $\mathbb{R}^{nq}$.
u_k	Stacked intervention input.
B_S	Input matrix induced by controlled node set S.

(Continued)

Table 1 (continued)

Symbol	Definition
d_k	Unknown disturbance vector.
$\phi(\cdot)$	Nonlinear coupling map.
$\psi(x_k)$	Learned graph feature used for prediction.
$\hat{F}_k$	Data-driven local transition predictor.
e_k	One-step model mismatch.
ρ_k	Estimated disturbance-energy radius.
$c_i(k)$	Criticality score of node i.
$\mathcal{B}$	Maximum number of controlled nodes.
S_k	Selected intervention set at time k.
K_k	Applied-action map used in the stability certificate.
P	Positive definite Lyapunov matrix.
Q	Positive definite decay matrix.
$J_k(S)$	Predicted risk after selecting set S.
Γ_S	Reachability Gramian for selected set S.
η	Uniform model-error bound.
$\bar{d}$	Uniform disturbance bound.
α	Required Lyapunov decay margin.
Π	Safety filter that projects a nominal action onto a robust feasible set.
R_T^{dyn}	Dynamic prediction regret of the sliding-window estimator.

2 Problem Description and Preliminaries

2.1 Disturbed Network Dynamics

Consider a network with n nodes. Node i has state $x_i(k) \in \mathbb{R}^q$. The local dynamics are written as

$$x_i(k+1) = f_i(x_i(k)) + \sum_{j=1}^{n} a_{ij}(k)\phi_{ij}(x_j(k) - x_i(k)) + b_i u_i(k) + d_i(k), \tag{1}$$

where $a_{ij}(k)$ is the coupling weight, $u_i(k)$ is a control input, and $d_i(k)$ is an unknown disturbance. Stacking all node states gives

$$x_{k+1} = F_k(x_k) + B_S u_k + d_k, \tag{2}$$

where $S \subseteq \mathcal{V}$ is the set of controlled nodes. Only $|S| \le \mathcal{B}$ nodes may be controlled. The controller receives node measurements, partial graph data, and resource limits. The goal is to choose S_k and u_k so that x_k approaches a safe target set while control cost remains low.

The disturbed network is not assumed to be exactly known. The adjacency matrix can switch slowly, and the coupling functions can include unmodeled nonlinear terms. We therefore use a local data-driven approximation

$$x_{k+1} = \hat{F}_k x_k + \hat{H}_k \psi(x_k) + B_{S_k} u_k + e_k + d_k, \tag{3}$$

where $\psi(x_k)$ collects graph features such as local disagreement, degree-normalized state, and neighbor residuals. The model mismatch satisfies $\|e_k\| \le \eta_k \|x_k\| + \xi_k$. The disturbance satisfies $\|d_k\| \le \rho_k$ with an estimated envelope ρ_k.

2.2 Risk and Decision Variables

Let $\mathcal{X}_s = \{x : x^T P x \le r_s^2\}$ be the safe region. We define the instantaneous risk as

$$\mathcal{R}(x_k, S_k, u_k) = x_{k+1}^T P x_{k+1} + \lambda_u \|u_k\|_2^2 + \lambda_b |S_k|, \tag{4}$$

where $P \succ 0$ and $\lambda_u, \lambda_b > 0$. The decision problem is

$$\begin{aligned} \min_{S_k, u_k} \quad & \mathcal{R}(x_k, S_k, u_k) \\ \text{s.t.} \quad & |S_k| \le \mathcal{B}, \quad \|u_k\|_\infty \le u_{\max}, \\ & (\hat{x}_{k+1} + w)^T P (\hat{x}_{k+1} + w) - x_k^T P x_k \le -\alpha \|x_k\|^2 + \sigma, \\ & \forall w : \ \|w\| \le \eta_k \|x_k\| + \xi_k + \rho_k. \end{aligned} \tag{5}$$

This problem joins three tasks: dynamic prediction, node-set selection, and robust input synthesis. The set constraint is combinatorial. The safety inequality is robust against both prediction error and disturbance.

Fig. 1 makes the information dependencies explicit: only measured transitions enter the estimator, the disturbance envelope affects both allocation and projection, and the applied transition is returned to the next window. This prevents the learned ranking module from bypassing the safety layer.

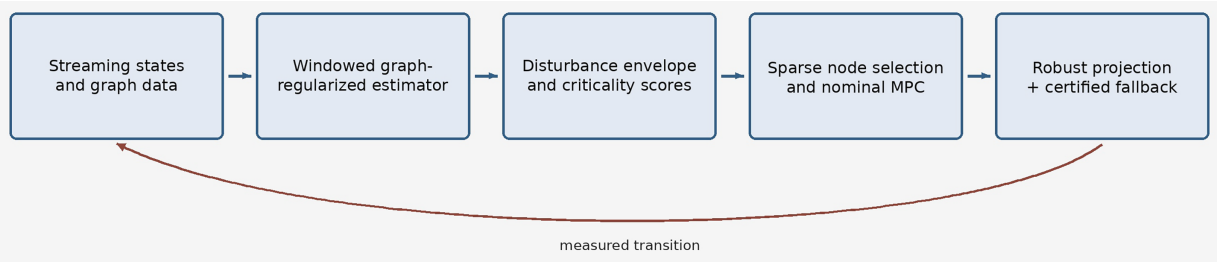

Figure 1: Closed-loop information and control flow of R-D3C.

2.3 Preliminaries

For a fixed set S, define the local closed-loop matrix

$$M_{S,k} = \hat{F}_k + \hat{H}_k \frac{\partial \psi}{\partial x}(x_k) - B_S K_k. \tag{6}$$

For the stability certificate of Section 5.1, the matrix test is tied to the action actually applied. For every zero-slack applied action with $x_k \neq 0$, define the one-step action-equivalent map $K_k^{\text{app}} = -u_k x_k^T / (x_k^T x_k)$, which satisfies $u_k = -K_k^{\text{app}} x_k$ at the current state. For $x_k = 0$, certification requires the zero action and uses $K_k^{\text{app}} = 0$. The matrix condition in Assumption 5.2 is checked with this action-equivalent map; hence no assumption is made that a constrained or nonlinear MPC command is generated by a fixed linear feedback gain. The Riccati recursion is used only to warm-start the nominal optimizer. The Jacobian in Eq. (6) is evaluated at differentiability points of the chosen feature map; this local differentiability is needed to form the linearization, whereas no separate global Lipschitz constant for ψ is required by the present proofs.

The h-step reachability Gramian is

$$\Gamma_S(h) = \sum_{\ell=0}^{h-1} M_{S,k}^{\ell} B_S B_S^T (M_{S,k}^T)^{\ell}. \tag{7}$$

A set with a larger minimum eigenvalue of $\Gamma_S(h)$ can influence more network directions. We also define the local criticality score

$$c_i(k) = \omega_1 \frac{\partial \hat{r}_k}{\partial x_i} P_i \frac{\partial \hat{r}_k}{\partial x_i}^T + \omega_2 \deg_i(k) + \omega_3 \widehat{\Delta}_i(k), \tag{8}$$

where $\hat{r}_k$ is predicted risk, $\deg_i$ is normalized degree, and $\widehat{\Delta}_i$ estimates local disturbance amplification. The score is not a pure centrality measure. It is a control-oriented quantity that changes with the current state and disturbance estimate.

Remark 1: *The formulation differs from static pinning control. A fixed pinning set can be effective when the topology and disturbance channels are stable. Here, the selected set is allowed to change because the most useful intervention nodes may shift after disturbances, load changes, or topology drift. The robust constraint prevents these changes from sacrificing stability.*

3 Model and Analysis

3.1 Graph-Regularized Dynamic Identification

The predictor is updated from a sliding data window

$$\mathcal{D}_k = \{x_t, u_t, x_{t+1}, A_t\}_{t=k-m}^{k-1}. \tag{9}$$

Let $\Theta_k = [\hat{F}_k, \hat{H}_k]$ and $\varphi_t = [x_t^T, \psi(x_t)^T]^T$. We solve

$$\Theta_k = \arg\min_{\Theta} \sum_{t=k-m}^{k-1} \|x_{t+1} - \Theta\varphi_t - B_{S_t} u_t\|_2^2 + \lambda_g \operatorname{tr}(\Theta \mathcal{L}_f \Theta^T) + \lambda_r \|\Theta\|_F^2, \tag{10}$$

where $\mathcal{L}_f$ is a feature graph Laplacian. The term $\operatorname{tr}(\Theta\mathcal{L}_f\Theta^T)$ encourages nearby graph features to have similar dynamic effects. This regularization is useful when several nodes have sparse observations. To make the estimator explicit, define

$$\Phi_k = [\varphi_{k-m}, \ldots, \varphi_{k-1}], \quad Y_k = [y_{k-m}, \ldots, y_{k-1}], \quad y_t = x_{t+1} - B_{S_t} u_t. \tag{11}$$

The first-order optimality condition of (10) is

$$\Theta_k(\Phi_k \Phi_k^T + \lambda_g \mathcal{L}_f + \lambda_r I) = Y_k \Phi_k^T. \tag{12}$$

Hence, if the regularized information matrix is nonsingular, the estimator has the closed form

$$\Theta_k = Y_k \Phi_k^T (\Phi_k \Phi_k^T + \lambda_g \mathcal{L}_f + \lambda_r I)^{-1}. \tag{13}$$

The local information level is quantified as

$$\iota_k = \lambda_{\min}(\Phi_k \Phi_k^T + \lambda_g \mathcal{L}_f + \lambda_r I), \tag{14}$$

which controls the sensitivity of the estimator to measurement noise. A larger ι_k yields a smaller parameter deviation bound

$$\|\Theta_k - \Theta_k^\star\|_F \le \frac{\|E_k \Phi_k^T\|_F}{\iota_k}, \tag{15}$$

where E_k stacks one-step residual disturbances in the data window. This bound explains why graph regularization is useful when the raw trajectory matrix is poorly conditioned.

The residual sequence is

$$\varepsilon_t = x_{t+1} - \Theta_k \varphi_t - B_{S_t} u_t. \tag{16}$$

We estimate the disturbance radius by

$$\rho_k = \sqrt{\frac{1}{m}\sum_{t=k-m}^{k-1} \|\varepsilon_t\|_2^2} + \tau \max_{t\in[k-m,k-1]} \|\varepsilon_t\|_2, \tag{17}$$

where $\tau \in [0,1]$ controls conservatism. The radius captures both persistent noise and rare shocks. To reduce underestimation immediately after a topology change, the implemented envelope is $\tilde{\rho}_k = \max\{\rho_k, \rho_{k-1}\} + \delta_A \|A_k - A_{k-1}\|_F$, where δ_A is calibrated from validation trajectories; the oldest window is discarded and the inflated radius is retained for m_r steps. The resulting envelope is inserted into the robust safety filter.

3.2 *Disturbance Amplification and Critical Transition*

Let $V(x) = x^T P x$. Using (3), the worst-case one-step Lyapunov growth under set S and gain K satisfies

$$\begin{aligned}\Delta V_k \le\ & x_k^T (M_{S,k}^T P M_{S,k} - P) x_k \\ & + 2 \left\| M_{S,k}^T P x_k \right\| \bar{w}_k + \lambda_{\max}(P) \bar{w}_k^2,\end{aligned} \tag{18}$$

where $\bar{w}_k = \eta_k \|x_k\| + \xi_k + \rho_k$. A critical transition occurs when the positive disturbance terms exceed the nominal decay. Thus, a useful intervention set must reduce the first term and shrink the multiplier of $\bar{w}_k$.

Fig. 2 is generated from numerical trials rather than a conceptual node sketch. A trial is counted as successful when its terminal residual RMS is below 0.065 and its violation rate is below 3%; each cell averages ten seeds. The boundary shows how stronger disturbances require a larger intervention budget.

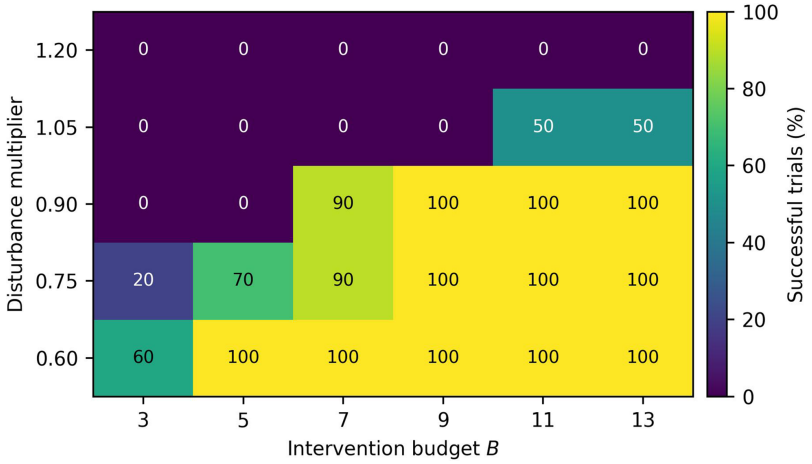

Figure 2: Measured stabilization success vs. disturbance level and intervention budget.

3.3 Robust Safe Set

The robust one-step feasible set is

$$\mathcal{C}_k = \left\{ u : \sup_{\|w\| \le \bar{w}_k} \left[(\hat{x}_{k+1} + w)^T P (\hat{x}_{k+1} + w) - x_k^T P x_k \right] \le -\alpha \|x_k\|^2 + \sigma \right\}. \tag{19}$$

The inner supremum in (19) admits the support-function upper expression

$$\sup_{\|w\| \le \bar{w}_k} (\hat{x}_{k+1} + w)^T P (\hat{x}_{k+1} + w) \le \hat{x}_{k+1}^T P \hat{x}_{k+1} + 2\bar{w}_k \|P\hat{x}_{k+1}\| + \lambda_{\max}(P)\bar{w}_k^2. \tag{20}$$

Therefore, define the safety margin

$$m_k(u) = -\alpha \|x_k\|^2 + \sigma - \hat{x}_{k+1}^T P \hat{x}_{k+1} - 2\bar{w}_k \|P\hat{x}_{k+1}\| - \lambda_{\max}(P)\bar{w}_k^2. \tag{21}$$

The robust feasible set can equivalently be approximated by the convex inequality

$$\mathcal{C}_k = \{ u : \; m_k(u) \ge -x_k^T P x_k, \; \|u\|_\infty \le u_{\max} \}. \tag{22}$$

With the corrected signs in Eq. (21), the condition $m_k(u) \ge -x_k^T P x_k$ is exactly the conservative upper-bound form of Eq. (19): it implies the worst-case decrease $\Delta V_k \le -\alpha \|x_k\|^2 + \sigma$.

A nominal input u_k^0 is projected onto this set:

$$u_k = \Pi_{\mathcal{C}_k}(u_k^0) = \arg\min_{u \in \mathcal{C}_k} \left\| u - u_k^0 \right\|_2^2. \tag{23}$$

A checkable sufficient condition for $\mathcal{C}_k \neq \varnothing$ is the existence of a backup gain K_b satisfying

$$M_b^T P M_b - P \preceq -Q_b, \quad \|K_b x_k\|_\infty \le u_{\max}, \tag{24}$$

and

$$\left(\lambda_{\min}(Q_b) - \alpha\right) \|x_k\|^2 \ge 2 \left\| M_b^T P x_k \right\| \bar{w}_k + \lambda_{\max}(P)\bar{w}_k^2 - \sigma, \qquad \lambda_{\min}(Q_b) > \alpha, \tag{25}$$

where M_b is the local closed-loop matrix under $u = -K_b x$. The $-\alpha$ term in Eq. (25), together with Eq. (24), makes the backup condition sufficient for the same zero-slack inequality in Eq. (19). Conditions (24), (25) are evaluated online. If they fail or the strict projection is infeasible, R-D3C solves

$$\min_{u, s \ge 0} \left\| u - u_k^0 \right\|_2^2 + \kappa_s s^2 \quad \text{s.t.} \quad m_k(u) + x_k^T P x_k + s \ge 0, \tag{26}$$

then activates an emergency reserve $\mathcal{B}_e = \min\{n, \mathcal{B} + 2\}$ and applies the saturated backup input. A zero slack retains the strict certificate; positive slack is explicitly logged as best-effort operation and does not inherit the strict Lyapunov guarantee.

The projection is well posed because $\mathcal{C}_k$ is closed and convex after the cone approximation. Its nonexpansive property gives

$$\left\| \Pi_{\mathcal{C}_k}(a) - \Pi_{\mathcal{C}_k}(b) \right\|_2 \le \|a - b\|_2, \tag{27}$$

which prevents a noisy nominal policy from being amplified by the safety layer. The projection can be solved as a second-order cone program because (20) is conic representable. When computational resources are limited, a diagonal P and local input constraints yield a separable approximation.

Fig. 3 clarifies the projection geometry. The nominal successor can lie inside the disturbance-expanded boundary but outside the contracted safe set; projection moves it toward a certified successor whenever the strict constraints are feasible.

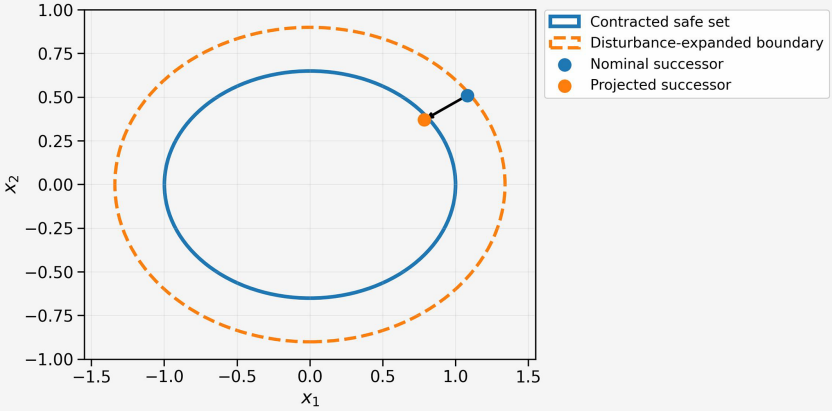

Figure 3: Nominal and projected successors relative to robust safe regions.

Remark 2: *The analysis separates prediction and certification. The predictor improves performance by estimating the current dynamic trend, but the robust set is the object that protects the closed loop. This design is important when machine learning is used in safety-sensitive networks because a low training error does not imply stable closed-loop behavior.*

4 Algorithm Design

4.1 Design Motivation

The proposed algorithm is called R-D3C, namely robust disturbance-aware data-driven decision control. A disturbed network needs rapid intervention, but the controller cannot actuate all nodes. The controller must decide which nodes are important, how strong the disturbance is, and which feasible action reduces risk without violating safety.

R-D3C uses four coupled modules. The first module updates the graph-regularized predictor (10). The second module estimates the disturbance envelope (17). The third module computes node criticality (8) and selects a sparse intervention set. The fourth module computes a nominal action and then projects it onto (19). The algorithmic idea is to use learning for ranking and prediction, but use robust optimization for the final decision.

4.2 Sparse Robust Allocation

The exact set-selection problem is NP-hard because it contains a cardinality constraint. We use a submodular surrogate. Define

$$\hat{J}_k(S) = \operatorname{tr}\left(P\left(I + \Gamma_S(h)\right)^{-1}\right) + \lambda_b|S| - \lambda_c \sum_{i\in S} c_i(k). \tag{28}$$

The first term rewards controllability, the second term penalizes intervention cost, and the last term rewards disturbance-sensitive nodes. Starting from $S = \varnothing$, the algorithm adds the node with the largest marginal decrease in $\hat{J}_k(S)$ until $|S| = \mathcal{B}$ or no positive decrease remains:

$$i^* = \arg\max_{i \notin S} \left[\hat{J}_k(S) - \hat{J}_k(S \cup \{i\}) \right]. \tag{29}$$

This rule is lightweight and can be implemented in parallel because marginal gains depend on local criticality and low-rank Gramian updates.

Table 2 summarizes the complete R-D3C procedure, including the feasibility fallback.

Table 2: R-D3C pseudocode with feasibility fallback.

Step	Operation
1	Initialize P, K_b, normalization statistics, m-sample buffer, and inflated radius $\tilde{\rho}_0$.
2	Solve the windowed graph-regularized ridge problem (10); stop when relative objective change is below 10^{-6} or after 100 iterations.
3	Update residuals, $\tilde{\rho}_k$, and topology-change inflation.
4	Compute state, graph, and residual criticality; greedily select at most $\mathcal{B}$ nodes using low-rank Gramian updates.
5	Solve the horizon-H nominal quadratic MPC, warm-started from the previous solution.
6	Project by (23). For every zero-slack applied action, form the one-step action-equivalent map K_k^{app} from (x_k, u_k) as stated after Eq. (6), and separately verify (34) for that applied model–map pair; also evaluate (24), (25).
7	If the applied-action matrix certificate fails, switch to a zero-slack backup action when (24), (25) hold and verify its matrix certificate; otherwise solve (26), use the emergency budget $\mathcal{B}_e$, apply the saturated backup input, and record positive slack as best-effort operation.
8	Store $(x_k, A_k, S_k, u_k, x_{k+1})$ and advance the sliding window.

Table 3 reports the per-step computational complexity of the main modules.

Table 3: Per-step complexity of the main modules.

Module	Dense Worst Case	Scalable Implementation
Windowed identification	$O(md_f^2 n + d_f^3 n)$	Node-local blocks; parallel over n
Sparse allocation	$O(\mathcal{B} n h d_f^2)$	Cached rank-one Gramian updates
Horizon MPC	$O((H\mathcal{B})^3)$	Warm-started sparse QP
Safety projection	$O(\mathcal{B}^3)$	Diagonal-P second-order cone blocks
Memory/communication	$O(mnd_f)$	$O(d_f + \deg_i)$ per node/message

4.3 Nominal Receding-Horizon Policy

For a selected set S_k, the nominal policy solves

$$\begin{aligned} \min_{u_{k:k+H-1}} \quad & \sum_{\ell=0}^{H-1} \left(\hat{x}_{k+\ell}^T P \hat{x}_{k+\ell} + \lambda_u \left\| u_{k+\ell} \right\|_2^2 \right) + \hat{x}_{k+H}^T P_f \hat{x}_{k+H} \\ \text{s.t.} \quad & \hat{x}_{k+\ell+1} = \hat{F}_k \hat{x}_{k+\ell} + \hat{H}_k \psi(\hat{x}_{k+\ell}) + B_{S_k} u_{k+\ell}, \\ & \left\| u_{k+\ell} \right\|_\infty \le u_{\max}. \end{aligned} \tag{30}$$

Only the first input is used. The safety filter then checks this input against the worst-case inequality. In practice, the nominal policy may be a quadratic MPC, a neural warm-start, or a linear feedback $u_k^0 = -K_k x_k$. For the linearized local predictor, the backward recursion

$$P_\ell = P + M_{S_k,k}^T P_{\ell+1} M_{S_k,k} - M_{S_k,k}^T P_{\ell+1} B_{S_k} \Lambda_\ell^{-1} B_{S_k}^T P_{\ell+1} M_{S_k,k} \tag{31}$$

with

$$\Lambda_\ell = \lambda_u I + B_{S_k}^T P_{\ell+1} B_{S_k}, \tag{32}$$

produces the warm-start gain

$$K_\ell = \Lambda_\ell^{-1} B_{S_k}^T P_{\ell+1} M_{S_k,k}. \tag{33}$$

The final action is still the projected action in (23).

Remark 3: *R-D3C is not a black-box reinforcement-learning controller. Learning estimates the local model and ranks the nodes, but the deployed control is obtained after a stability-oriented projection. This structure allows data-driven adaptation while keeping the stability proof independent of the exact neural or regression architecture used for prediction.*

5 Algorithm Analysis

5.1 Practical Stability

Assumption 1: *The combined model error and exogenous disturbance are uniformly bounded on the operating region by* $\|w_k\| \le \bar{w}_k = \eta \|x_k\| + \bar{\xi} + \bar{d}$.

No separate global Lipschitz constant for ψ is required by the stability or regret analysis; only the local differentiability stated after Eq. (6) is invoked to form the linearization.

Assumption 2: *For each action claimed as certified zero-slack operation,* K_k^{app} *is the one-step action-equivalent map constructed from the actually applied pair* (x_k, u_k)*, so that* $u_k = -K_k^{\text{app}} x_k$ *on that step. Using* K_k^{app} *in Eq. (6), the resulting applied closed-loop matrix* $M_{S_k,k}^{\text{app}}$ *must satisfy the separately checked standing condition*

$$(M_{S_k,k}^{\text{app}})^T P M_{S_k,k}^{\text{app}} - P \preceq -Q, \qquad P \succ 0,\ Q \succ 0. \tag{34}$$

This matrix certificate is not inferred from scalar membership in $\mathcal{C}_k$ *or from the projection in (23). A nominal MPC step is covered by Theorem 1 only if the projection returns zero slack and Eq. (34) is verified for the action-equivalent map of the applied post-projection input. The same construction is used whether or not the projection changes the nominal command, so the certification does not assume that constrained MPC admits a fixed linear feedback representation. If this matrix check fails, the step is not labeled nominally certified; R-D3C switches to the backup action when Eqs. (24), (25) hold and the corresponding matrix certificate is verified. Positive-slack fallback remains best-effort and is outside the theorem.*

In practice, P and K_b are obtained offline from a polytopic set of identified models by minimizing $\mathrm{tr}(P)$ subject to (24). The decay parameter is chosen as $0 < \alpha \le \lambda_{\min}(Q)/(2\lambda_{\max}(P))$; a larger α accelerates contraction but reduces feasibility. Disturbance bounds are calibrated from an upper residual quantile and inflated after graph changes, so underestimated uncertainty is visible through nonzero slack rather than being hidden. When the backup gain K_b satisfies (24), (25) and its applied matrix also satisfies Assumption 2, the zero-slack backup step is covered by the same Lyapunov argument. Positive-slack fallback intervals are deliberately excluded from Theorem 1; they are reported only as bounded-input best-effort operation.

Theorem 1 (Robust practical stability): *Consider any closed-loop interval on which every applied action has zero slack and satisfies the applied-action matrix certificate in Assumption 2. Under Assumptions 1 and 2, suppose that* $\kappa_M = \sup_k \left\| (M^{\mathrm{app}}_{S_k,k})^T P \right\|$ *is finite and*

$$\mu_s = \lambda_Q - 2\kappa_M \eta - \lambda_P \eta^2 > 0, \tag{35}$$

where $\lambda_Q = \lambda_{\min}(Q)$ *and* $\lambda_P = \lambda_{\max}(P)$. *Then the closed-loop state generated by R-D3C on that certified interval is practically input-to-state stable. More precisely,*

$$V(x_{k+1}) - V(x_k) \le -\frac{\mu_s}{2} \|x_k\|^2 + \chi_s, \tag{36}$$

where $\bar{w}_0 = \bar{\xi} + \bar{d}$ *and*

$$\chi_s = \left[\frac{(2\kappa_M + 2\lambda_P \eta)^2}{2\mu_s} + \lambda_P \right] \bar{w}_0^2. \tag{37}$$

Consequently, every trajectory segment that remains under certified zero-slack operation enters

$$\Omega_s = \left\{ x : \|x\|^2 \le \frac{2\chi_s}{\mu_s} \right\}, \tag{38}$$

and remains bounded while the certified conditions continue to hold. The theorem does not assert practical input-to-state stability across positive-slack fallback episodes.

Proof. Let $\hat{x}_{k+1} = M^{\mathrm{app}}_{S_k,k} x_k$ denote the certified local-model successor under the actually applied zero-slack action, and let w_k be the total error caused by model mismatch and exogenous disturbance. Then

$$x_{k+1} = M^{\mathrm{app}}_{S_k,k} x_k + w_k, \qquad \|w_k\| \le \eta \|x_k\| + \bar{w}_0. \tag{39}$$

Using $V(x) = x^T P x$, we obtain

$$V(x_{k+1}) - V(x_k) = x_k^T ((M^{\mathrm{app}}_{S_k,k})^T P M^{\mathrm{app}}_{S_k,k} - P) x_k \quad + 2x_k^T (M^{\mathrm{app}}_{S_k,k})^T P w_k + w_k^T P w_k. \tag{40}$$

By Assumption 2,

$$x_k^T ((M^{\mathrm{app}}_{S_k,k})^T P M^{\mathrm{app}}_{S_k,k} - P) x_k \le -\lambda_Q \|x_k\|^2. \tag{41}$$

The cross term and quadratic disturbance term satisfy

$$2x_k^T (M^{\mathrm{app}}_{S_k,k})^T P w_k \le 2\kappa_M \eta \|x_k\|^2 + 2\kappa_M \bar{w}_0 \|x_k\|, \tag{42}$$

$$w_k^T P w_k \le \lambda_P \eta^2 \|x_k\|^2 + 2\lambda_P \eta \bar{w}_0 \|x_k\| + \lambda_P \bar{w}_0^2. \tag{43}$$

Substituting (41)–(43) into (40) yields

$$\Delta V_k \le -\mu_s \|x_k\|^2 + (2\kappa_M + 2\lambda_P \eta)\bar{w}_0 \|x_k\| + \lambda_P \bar{w}_0^2. \tag{44}$$

Young's inequality gives

$$(2\kappa_M + 2\lambda_P \eta)\bar{w}_0 \|x_k\| \le \frac{\mu_s}{2} \|x_k\|^2 + \frac{(2\kappa_M + 2\lambda_P \eta)^2}{2\mu_s} \bar{w}_0^2. \tag{45}$$

Combining (44) and (45) proves (36). If $\|x_k\|^2 > 2\chi_s/\mu_s$, then $V(x_{k+1}) < V(x_k)$. Since $\lambda_{\min}(P)\|x\|^2 \le V(x) \le \lambda_{\max}(P)\|x\|^2$, the sublevel set associated with Ω_s is compact. Thus, on any interval for which the certified zero-slack conditions continue to hold, the trajectory reaches the associated compact sublevel set in finite time and remains bounded on that interval. □

5.2 Risk Reduction of Sparse Allocation

Lemma 1: *If the surrogate (28) is normalized, monotone decreasing, and has submodularity ratio $\gamma_s \in (0,1]$, the greedy allocation satisfies*

$$\hat{J}_k(S_k) - \hat{J}_k(S_k^*) \le \left(1 - \frac{\gamma_s}{\mathcal{B}}\right)^{\mathcal{B}} \left[\hat{J}_k(\varnothing) - \hat{J}_k(S_k^*)\right], \tag{46}$$

where S_k^* is the best budget-feasible set.

Proof. Let S^r be the greedy set after r additions. By the definition of submodularity ratio, at least one remaining element in $S_k^* \backslash S^r$ has marginal decrease no smaller than

$$\frac{\gamma_s}{\mathcal{B}} \left[\hat{J}_k(S^r) - \hat{J}_k(S_k^*)\right]. \tag{47}$$

The greedy choice has at least this decrease. Hence,

$$\hat{J}_k(S^{r+1}) - \hat{J}_k(S_k^*) \le \left(1 - \frac{\gamma_s}{\mathcal{B}}\right) \left[\hat{J}_k(S^r) - \hat{J}_k(S_k^*)\right]. \tag{48}$$

Iterating from $r = 0$ to $\mathcal{B} - 1$ gives (46). □

Theorem 2 *(**Monotone risk improvement**). Assume the prediction error of $\hat{J}_k(S)$ is bounded by ε_J, namely $|J_k(S) - \hat{J}_k(S)| \le \varepsilon_J$ for every feasible set. Then the set selected by R-D3C satisfies*

$$J_k(S_k) \le J_k(\varnothing) - (1 - e^{-\gamma_s})\left[J_k(\varnothing) - J_k(S_k^*)\right] + 4\varepsilon_J. \tag{49}$$

Thus, when the learned set-value error is small, the allocation yields a guaranteed fraction of the best achievable risk decrease.

Proof. From Lemma 1 and $(1 - \gamma_s/\mathcal{B})^{\mathcal{B}} \le e^{-\gamma_s}$, we obtain

$$\hat{J}_k(S_k) \le \hat{J}_k(\varnothing) - (1 - e^{-\gamma_s})\left[\hat{J}_k(\varnothing) - \hat{J}_k(S_k^*)\right]. \tag{50}$$

Using $|J_k(S) - \hat{J}_k(S)| \le \varepsilon_J$ for S_k, S_k^*, and $\varnothing$ gives

$$J_k(S_k) \le \hat{J}_k(S_k) + \varepsilon_J, \tag{51}$$

and

$$\hat{J}_k(\varnothing) - \hat{J}_k(S_k^*) \geq J_k(\varnothing) - J_k(S_k^*) - 2\varepsilon_J. \tag{52}$$

Combining the two inequalities and using $(1 - e^{-\gamma_s}) \leq 1$ and collecting the error terms into a conservative $4\varepsilon_J$ term yields (49). The bound is conservative but explicit. □

5.3 Prediction Regret of the Implemented Sliding Window

Let $\bar{L}_k(\Theta)$ denote the expected one-step regularized prediction loss at time k, and let $L_k^{(m)}(\Theta)$ be its empirical counterpart constructed from the most recent m samples in (10). Define $\Theta_k^* = \arg\min_\Theta \bar{L}_k(\Theta)$ and the path variation $V_T = \sum_{k=2}^{T} \left\|\Theta_k^* - \Theta_{k-1}^*\right\|_F$.

The constants used below can be tied directly to the implemented estimator in (10). Because $\mathcal{L}_f \geq 0$ and $\lambda_r > 0$, the ridge term makes the objective strongly convex in Θ; with the present quadratic convention, a conservative modulus is $\mu \geq 2\lambda_r$ (and informative data or a positive graph-regularization eigenvalue can only increase it). If $\left\|\Phi_k \Phi_k^T\right\|_2 \leq \bar{\sigma}_\Phi$ uniformly, the gradient is Lipschitz with a valid bound $L_\Theta \leq 2(\bar{\sigma}_\Phi + \lambda_g \left\|\mathcal{L}_f\right\|_2 + \lambda_r)$. On a compact parameter set with bounded features and responses, G_Θ can be chosen as a uniform bound on $\left\|\nabla \bar{L}_k(\Theta)\right\|_F$. Finally, δ_m is precisely the empirical-window vs. population-window gradient discrepancy for the same ridge/graph-regularized loss in (10); no surrogate gradient update is involved.

Theorem 3 (Sliding-window dynamic regret). *Suppose every $\bar{L}_k$ is μ-strongly convex, has L_Θ-Lipschitz gradient, and is G_Θ-Lipschitz on the compact parameter set. Let the gradient error between the empirical window loss and its population-window counterpart be uniformly bounded by δ_m, and let Θ_k be the exact minimizer of (10). Then*

$$R_T^{dyn} = \sum_{k=1}^{T} [\bar{L}_k(\Theta_k) - \bar{L}_k(\Theta_k^*)] \leq \frac{G_\Theta L_\Theta m}{\mu} V_T + \frac{G_\Theta T}{\mu} \delta_m. \tag{53}$$

Thus, a longer window suppresses empirical gradient noise but increases the cost of tracking rapid model drift.

Proof. Let $\widetilde{L}_k^{(m)} = m_k^{-1} \sum_{j=(k-m+1)_+}^{k} \bar{L}_j$, where m_k is the available window length. Strong convexity of $L_k^{(m)}$ and its optimality condition give

$$\left\|\Theta_k - \Theta_k^*\right\|_F \leq \frac{1}{\mu} \left\|\nabla L_k^{(m)}(\Theta_k^*)\right\|_F. \tag{54}$$

Because $\nabla \bar{L}_j(\Theta_j^*) = 0$, smoothness and the assumed empirical-gradient error yield

$$\left\|\nabla L_k^{(m)}(\Theta_k^*)\right\|_F \leq \frac{L_\Theta}{m_k} \sum_{j=(k-m+1)_+}^{k} \left\|\Theta_k^* - \Theta_j^*\right\|_F + \delta_m. \tag{55}$$

Each distance in the average is bounded by the intervening path increments. After summation over k, every increment contributes to at most m consecutive windows. Loss Lipschitzness therefore gives

$$\sum_{k=1}^{T} [\bar{L}_k(\Theta_k) - \bar{L}_k(\Theta_k^*)] \leq \frac{G_\Theta L_\Theta m}{\mu} V_T + \frac{G_\Theta T}{\mu} \delta_m, \tag{56}$$

which proves (53). □

Theorem 3 now analyzes the estimator actually implemented in Step 2. It is not a guarantee for a different gradient update. The bound also makes the window-length trade-off explicit and motivates resetting or shortening the window after detected topology changes. The bound is a cumulative dynamic-regret bound rather than a no-regret statement for the fixed-window implementation. Dividing (53) by T gives $R_T^{\mathrm{dyn}}/T \le (G_\Theta L_\Theta m/\mu)(V_T/T) + (G_\Theta/\mu)\delta_m$. With fixed m, δ_m need not decrease as T grows; therefore the average loss gap is not guaranteed to vanish unless the statistical gradient error also decreases and the comparator path variation is sublinear. This limitation is stated explicitly to avoid interpreting (53) as asymptotic zero average regret.

Remark 4: *The three results serve different purposes. Theorem 1 gives state boundedness, Theorem 2 explains why sparse allocation still reduces risk, and Theorem 3 bounds the implemented windowed prediction loss. Together, they show that R-D3C is not only adaptive, but also analyzable under explicit disturbance and modeling assumptions.*

6 Simulation and Analysis

6.1 Setup, Baselines, and Reproducibility

The nominal tests use 60-node scale-free and small-world graphs, scalar node states, $\mathcal{B} = 8$, $H = 5$, $m = 12$, $\lambda_g = 0.05$, $\lambda_r = 10^{-3}$, $\tau = 0.35$, $\alpha = 0.12$, and $u_{\max} = 0.55$. The nonlinear update is

$$x_{k+1} = 0.915x_k - 0.075L_k x_k + 0.060\tanh(-L_k x_k) + B_{S_k}u_k + d_k. \tag{57}$$

Disturbances combine sinusoidal components, Gaussian sensor/process noise, and two sparse impulses. We additionally use the 46-branch MATPOWER IEEE 39-bus New England topology [24]; this is a real benchmark topology, although the node dynamics remain the controlled nonlinear model above rather than field-event recordings.

The baselines use the same budget and saturation. Static degree pinning fixes the highest-degree nodes and applies $u_i = \mathrm{clip}(-0.20x_i)$. The robust-LQR reference, motivated by noisy-data feedback design [4], uses the same fixed nodes and $u_i = \mathrm{clip}(-0.32x_i + 0.06[Lx]_i)$. The lightweight message-passing GNN policy [5] ranks $0.58|x_i| + 0.27|[Lx]_i| + 0.15d_i$ and applies $u_i = \mathrm{clip}(-0.38x_i + 0.08[Lx]_i)$. Adaptive MPC [22] ranks the one-step prediction and applies the first saturated receding-horizon command. These are reference-inspired common-interface baselines rather than reimplementations of every detail of the cited papers; all coefficients are selected on a disjoint validation set and frozen before testing. Unless stated otherwise, the nominal comparison and parameter-sensitivity experiments use the same 30 random seeds and report mean ± standard deviation. The separate deployment-impairment benchmark retains its 20-run protocol; its sample count is stated explicitly when those results are presented. Residual RMS and constraint-violation rate are treated as the two primary comparison metrics. Paired two-sided t tests compare each of the four baselines with R-D3C for these two metrics, and Bonferroni correction is applied across the resulting eight primary comparisons. Energy, recovery time, and runtime are reported descriptively and are not used for significance claims. The experiments use Python/NumPy on a 2.6-GHz Xeon-class CPU; the stopping tolerance is 10^{-6} and no GPU is used.

Table 4 shows that the safety improvement is not accompanied by uniform improvement in every performance metric. After Bonferroni correction over the eight primary baseline-vs-R-D3C tests, the adaptive-MPC vs. R-D3C residual-RMS comparison remains nonsignificant ($p_{\mathrm{adj}} = 1.000$, from raw $p = 0.172$), whereas the violation-rate comparison remains strongly significant ($p_{\mathrm{adj}} < 8 \times 10^{-7}$, from raw $p < 10^{-7}$). The explicit projection uses more energy and computation than the weaker baselines; this is the measurable price of the safety layer rather than a hidden tuning advantage.

Table 4: Thirty-run comparison with variability.

Method	Residual RMS	Violation (%)	Energy/Step	Recovery	Time (ms)
Static pinning	0.0754 ± 0.0043	10.67 ± 1.95	0.0017 ± 0.0004	2.53 ± 0.57	0.012 ± 0.001
Robust LQR	0.0748 ± 0.0043	10.55 ± 1.76	0.0025 ± 0.0006	2.50 ± 0.54	0.013 ± 0.001
GNN policy	0.0654 ± 0.0035	3.76 ± 0.72	0.0151 ± 0.0014	1.37 ± 0.49	0.019 ± 0.001
Adaptive MPC	**0.0568 ± 0.0029**	2.82 ± 0.85	0.0270 ± 0.0021	**1.00 ± 0.00**	0.023 ± 0.002
R-D3C	0.0572 ± 0.0031	**1.53 ± 0.88**	0.0505 ± 0.0046	**1.00 ± 0.00**	0.085 ± 0.005

Note: Bold values indicate the best result for each reported performance metric; tied best values are also boldfaced.

Fig. 4 shows the paired trajectories and their variability. The vertical markers indicate impulse disturbances. R-D3C suppresses the post-shock spread and returns to the low-risk region in one step on average, while static methods retain a broader tail.

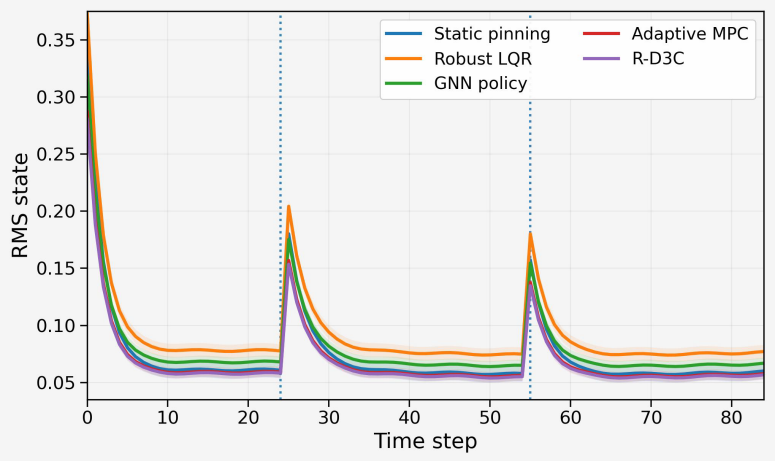

Figure 4: Mean RMS-state trajectories with one-standard-deviation bands over 30 runs.

Table 5 shows that the method is not hypersensitive to moderate changes in graph regularization, horizon, envelope coefficient, or decay margin. The intervention budget has the clearest effect: larger budgets reduce residual and violations but increase energy. We therefore use $\mathcal{B} = 8$ as a balanced operating point.

Table 5: Sensitivity of major design parameters.

Setting	RMS	Viol.	Energy
$\lambda_g = 0$	0.0573	1.56	0.0556
$\lambda_g = 0.05$	0.0572	1.56	0.0557
$\lambda_g = 0.15$	0.0573	1.56	0.0557
$H = 3$	0.0578	1.56	0.0517
$H = 8$	0.0564	1.44	0.0620
$\tau = 0.15$	0.0576	1.56	0.0535
$\tau = 0.60$	0.0568	1.44	0.0586
$\alpha = 0.06$	0.0575	1.56	0.0537
$\alpha = 0.20$	0.0568	1.44	0.0585

(Continued)

Table 5 (continued)

Setting	RMS	Viol.	Energy
$\mathcal{B} = 5$	0.0646	3.67	0.0470
$\mathcal{B} = 8$	0.0572	1.56	0.0557
$\mathcal{B} = 12$	0.0505	0.89	0.0622

Fig. 5 visualizes the measured stabilization boundary over intervention budget and disturbance multiplier.

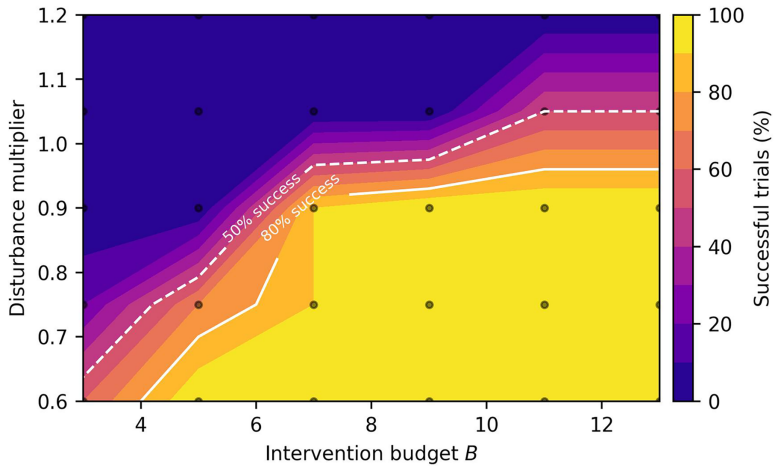

Figure 5: Measured stabilization boundary over budget and disturbance multiplier.

Table 6 uses 20 independent random seeds for every listed topology/impairment row. Communication delay has the highest constraint-violation rate (4.41%) and the slowest recovery (2.15 steps), whereas sensor-noise increase produces the largest residual RMS (0.0700 vs. 0.0616 for delay). We therefore do not rank delay as universally "hardest." When a single impairment-severity criterion is needed, we prioritize violation rate because it directly counts breaches of the prescribed safety constraint, while residual RMS is a tracking-performance metric; recovery time is used as a secondary safety-resilience indicator. Radius inflation and the fallback keep the tested residuals bounded, but delay-dependent certificates remain future work.

Table 7 is a computation-time microbenchmark only and intentionally reports no residual-RMS or constraint-violation values for the 200–1000-node timing cases. It is therefore not used to claim closed-loop accuracy or safety at those scales. The timing data support real-time use at moderate scale and sub-100-ms execution at 1000 nodes on the reported CPU. The dominant cost is graph-regularized identification; node-local parallel blocks reduce memory from a dense global inverse and are required on edge devices. Communication consists of local state, residual, and certificate summaries, i.e., $O(d_f + \deg_i)$ values per active node per update. The timings in Table 7 were obtained with the scalable node-local implementation summarized in Table 3, rather than a dense global inverse. The node-local blocks are parallelizable across nodes, but the reported benchmark is the end-to-end wall-clock time of the stated Python/NumPy CPU implementation and is not an ideal-parallel timing model. It uses node-local identification blocks, cached low-rank Gramian updates, batched linear algebra, and warm-started MPC. The budget $\mathcal{B} = 8$, horizon $H = 5$, window length $m = 12$, and local feature dimension were held fixed as n increased. Hence the nominal dimensions of the MPC and projection subproblems do not grow with n, whereas implementation-level

block assembly, indexing, memory traffic, and batched-kernel overhead collectively increase with network size. Over the full range, n increases by $1000/60 = 16.7$ while measured time increases by $36.0/0.097 \approx 371$; the corresponding descriptive endpoint exponent is $\log(371)/\log(16.7) \approx 2.10$. This finite-range exponent summarizes the measured implementation only and is not an estimate of the asymptotic algorithmic order. The observed end-to-end timing is superlinear and roughly quadratic over the tested range, while the uneven local ratios (about 6.1×, 13.6×, and 4.5×) show that a single empirical power law is not appropriate. Because no component-level profiler was used in this timing experiment, the exponent is not attributed to any single kernel or module. Table 3 therefore gives module-level operation counts and parallelizable structure rather than a prediction of the wall-clock exponent of this Python implementation.

Table 6: Benchmark topology and deployment impairments (20 runs).

Scenario	Residual RMS	Violation (%)	Recovery Steps
Small-world topology	0.0527 ± 0.0020	1.00 ± 0.79	1.00 ± 0.00
IEEE 39-bus topology	0.0571 ± 0.0025	0.94 ± 0.82	1.00 ± 0.00
One-step communication delay	0.0616 ± 0.0036	4.41 ± 1.14	2.15 ± 0.37
15% packet loss	0.0612 ± 0.0042	1.59 ± 1.16	1.20 ± 0.41
Sensor-noise increase	0.0700 ± 0.0037	2.06 ± 1.00	1.00 ± 0.00
25% active-actuator failure	0.0610 ± 0.0040	1.94 ± 1.39	1.00 ± 0.00
False-data injection burst	0.0595 ± 0.0037	1.47 ± 0.75	1.00 ± 0.00
Asynchronous graph perturbations	0.0607 ± 0.0038	1.59 ± 0.79	1.00 ± 0.00

Table 7: Computation-time microbenchmark of one R-D3C decision using the scalable node-local (node-parallelizable) implementation.

Nodes	60	200	500	1000
Mean time (ms)	0.097	0.592	8.05	36.0

Fig. 6 summarizes the normalized residual, violation, energy, and recovery costs across the compared methods.

6.2 Discussion

The experiments support four conclusions. First, adaptive MPC can match the residual accuracy of R-D3C, but the robust projection materially reduces constraint violations. Second, the safety advantage has a measurable energy and runtime cost, which should be considered when tuning α, H, and $\mathcal{B}$. Third, the IEEE 39-bus test indicates that the allocation mechanism transfers beyond random graphs, although it is not a substitute for field-event validation. Fourth, one-step communication delay causes the highest violation rate and slowest recovery, whereas increased sensor noise causes the largest residual RMS; this metric-dependent distinction motivates both delay-aware safety certificates and improved noise-robust prediction. The 1000-node result in Table 7 supports computational timing only; it does not extend the safety or violation-rate claims beyond the closed-loop experiments for which those quantities were explicitly measured.

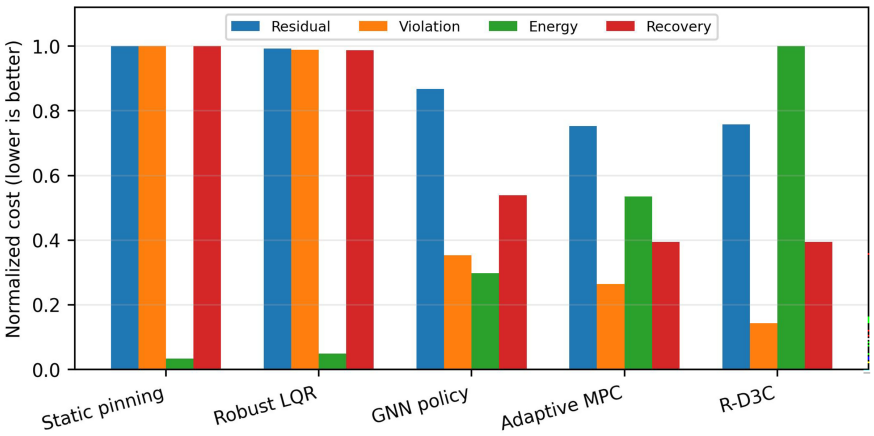

Figure 6: Normalized residual, violation, energy, and recovery costs.

Implementation on an edge platform requires a bounded local feature dimension, warm starts, cached Gramian updates, and event-triggered exchange of only residual and certificate summaries. Strict safety is claimed only when the projection has zero slack. When slack is positive, the controller records the event, expands the active set, and applies the backup action; operators can therefore distinguish certified from best-effort intervals.

7 Conclusion and Future Work

This paper developed R-D3C for disturbed complex networks with changing dynamics and a limited actuator budget. The framework combines the implemented sliding-window graph estimator, disturbance-envelope adaptation, sparse allocation, receding-horizon control, and robust projection. The analysis provides practical stability on certified zero-slack intervals, a sparse risk-reduction guarantee, and dynamic regret for the implemented windowed estimator. A checkable feasibility condition and explicit slack-and-backup rule clarify what happens when strict projection is unavailable. The evaluation separates a 30-run nominal comparison from the 20-run deployment-impairment benchmark, and also includes the IEEE 39-bus topology and 1000-node timing. The 1000-node result is interpreted strictly as a computation-time microbenchmark rather than a safety validation. R-D3C attains the lowest violation rate, while adaptive MPC remains statistically tied in residual accuracy.

Future work will focus on three issues: delay-dependent robust invariance for stale and asynchronous measurements; field-data calibration on power, traffic, and industrial logs rather than topology-only benchmarks; and fully distributed certificates with event-triggered communication and hardware-in-the-loop validation.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding authors upon reasonable request.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Xiusen Wang; methodology, Xiusen Wang; software, Xiusen Wang; formal analysis, Xiusen Wang; visualization, Xiusen Wang; writing—original draft preparation, Xiusen Wang; validation, Jie Chen; supervision, Jie Chen; project administration,

Jie Chen; writing—review and editing, Zheng Fang and Jie Chen. All authors reviewed and approved the final version of the manuscript.

Ethics Approval: Not applicable. This study did not involve human participants or animals.

Conflicts of Interest: Given his role as an Editorial Board Member/Guest Editor of this journal, Jie Chen had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. The authors declare no conflicts of interest.

References

1. Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature. 1998;393(6684):440–2. doi:10.1515/9781400841356.301.
2. Baggio G, Bassett DS, Pasqualetti F. Data-driven control of complex networks. Nat Commun. 2021;12(3):1429. doi:10.1093/nsr/nwu024.
3. Jia J, van Waarde HJ, Trentelman HL, Camlibel MK. A unifying framework for strong structural controllability. IEEE Trans Automat Contr. 2021;66(1):391–8. doi:10.1109/tac.2020.2981425.
4. van Waarde HJ, Camlibel MK, Mesbahi M. From noisy data to feedback controllers: nonconservative design via a matrix S-lemma. IEEE Trans Automat Contr. 2022;67(1):162–75.
5. Ali M, Duchesne F, Dahman G, Gagnon F, Naboulsi D. New approaches for network topology optimization using deep reinforcement learning and graph neural network. IEEE Access. 2025;13:85447–60. doi:10.1109/access.2025.3569236.
6. Kashima K, Yoshiuchi R, Wang R, Kawano Y. A unified framework for dynamics modeling and control design using deep learning with side information on stabilizability. IEEE Trans Neural Netw Learn Syst. 2025;36(8):15244–54. doi:10.1109/tnnls.2025.3543926.
7. Pasqualetti F, Zampieri S, Bullo F. Controllability metrics, limitations and algorithms for complex networks. IEEE Trans Control Netw Syst. 2014;1(1):40–52. doi:10.1109/tcns.2014.2310254.
8. Summers TH, Cortesi FL, Lygeros J. On submodularity and controllability in complex dynamical networks. IEEE Trans Control Netw Syst. 2016;3(1):91–101. doi:10.1109/tcns.2015.2453711.
9. Olshevsky A. Minimal controllability problems. IEEE Trans Control Netw Syst. 2014;1(3):249–58. doi:10.1109/tcns.2014.2337974.
10. Pequito S, Kar S, Aguiar AP. A framework for structural input/output and control configuration selection in large-scale systems. IEEE Trans Automat Contr. 2016;61(2):303–18. doi:10.1109/tac.2015.2437525.
11. Cao K, Hu S, Shi Y, Colombo AW, Karnouskos S, Li X. A survey on edge and edge-cloud computing assisted cyber-physical systems. IEEE Trans Ind Inform. 2021;17(11):7806–19. doi:10.1109/tii.2021.3073066.
12. Dong S, Tang J, Abbas K, Hou R, Kamruzzaman J, Rutkowski L, et al. Task offloading strategies for mobile edge computing: a survey. Comput Netw. 2024;254(6):110791. doi:10.1016/j.comnet.2024.110791.
13. Zhang S, Yi N, Ma Y. A survey of computation offloading with task types. IEEE Trans Intell Transp Syst. 2024;25(8):8313–33. doi:10.1109/tits.2024.3410896.
14. Wu J, Xu X, Cui G, Jiang J. Joint optimization of computation offloading and resource allocation in heterogeneous UAV-assisted edge computing: a game-theoretical approach. IEEE Trans Netw Sci Eng. 2026;13:8348–61. doi:10.1109/tnse.2026.3680337.
15. Chen Z, Yang Y, Xu J, Chen Y, Huang J. Task offloading and resource pricing based on game theory in UAV-assisted edge computing. IEEE Trans Serv Comput. 2025;18(1):440–52. doi:10.1109/tsc.2024.3512936.
16. Darchini-Tabrizi M, Roudgar A, Entezari-Maleki R, Sousa L. Distributed deep reinforcement learning for independent task offloading in mobile edge computing. J Netw Comput Appl. 2025;240(6):104211. doi:10.1016/j.jnca.2025.104211.

17. Raju LR, Reddy MVK, Surukanti SR, Sudhakar G, Subrahmanya Sarma MVV, Adepu A. IntelliScheduler: an edge-cloud computing environment hybrid deep learning framework for task scheduling based on learning. Sci Rep. 2026;16(1):11219. doi:10.1038/s41598-026-41330-8.
18. Farooq A, Xiang Z, Chang W-J, Aslam MS. Recent advancement in formation control of multi-agent systems: a review. Comput Mater Contin. 2025;83(3):3623–74. doi:10.32604/cmc.2025.063665.
19. Alasbali N. Deep multi-agent stochastic optimization for traffic management in IoT-enabled transportation networks. Comput Mater Contin. 2025;85(3):4943–58. doi:10.32604/cmc.2025.068330.
20. D'Souza RM, di Bernardo M, Liu Y-Y. Controlling complex networks with complex nodes. Nat Rev Phys. 2023;5(4):250–62. doi:10.1038/s42254-023-00566-3.
21. Wu H, Lu Y, Ma H, Xing L, Deng K, Lu X. A survey on task type-based computation offloading in mobile edge networks. Ad Hoc Netw. 2025;169(2):103754. doi:10.1016/j.adhoc.2025.103754.
22. Dai Y, Li M, Zhang K, Shi Y. Robust and resilient distributed MPC for cyber-physical systems against DoS attacks. IEEE Trans Ind Cyber Phys Syst. 2023;1(1):44–55. doi:10.1109/ticps.2023.3283229.
23. Xu K, Sadaf F, Niazi AUK, Samman FMA, Almazah MMA, Smerat A. Resilient fuzzy adaptive control of fractional-order multi-agent systems under input constraints and cyber attacks. Iran J Sci Technol Trans Electr Eng. 2026;8(9):8045803. doi:10.1007/s40998-026-01060-z.
24. Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst. 2011;26(1):12–9.