

Supplementary Information for

Physics-informed graph learning of collapse distance in complex networks

Jie Zhang^{1,#}, Tao Wang^{1,#}, Yatai Ji^{1,2}, Hua He¹, Zhengqiu Zhu^{1,2}, Boquan Zhang¹,
Xin Zhou¹, Changjun Fan¹, Bin Chen³, Manlio De Domenico^{4,*}, and Xin Lu^{1,*}

¹College of Systems Engineering, National University of Defense Technology, Changsha, 410073, China

²State Key Laboratory of Digital Intelligent Modeling and Simulation, National University of Defense
Technology, Changsha, 410073, China

³The Institute of Intelligent Computing, University of Electronic Science and Technology of China,
Chengdu 611731, China

⁴Department of Physics and Astronomy “Galileo Galilei”, University of Padua, Italy

#These authors contributed equally to this work.

*Correspondence: Manlio De Domenico (manlio.dedomenico@unipd.it), Xin Lu (xin.lu.lab@outlook.com).

CONTENTS

I	Mathematical Proofs of Collapse Distance Properties	6
A	Proof of Monotonicity and Smoothness	6
B	Proof of Scale-Transformed Linear Additivity	8
II	Theoretical Analysis of TCR-GIN Framework.....	10
A	Sufficient Condition for Surpassing Individual Heuristics under One-Sided Label Bias	10
B	Bounded True-Label Error and Monotone Contraction under Heuristic-Pool Expansion	13
C	Reducing Supervision Bias That Persists Regardless of Training-Set Size	19
D	Approximate Additivity of TCR-GIN Predictions	24
E	Computational Complexity Analysis	27
III	Robustness Metrics	28
A	Definition	28
B	Detailed Description of Main Metrics	29
IV	Baselines	31
A	Centrality-based Methods	31
B	Mathematical Programming & Spectral Analysis	33
C	Heuristics Methods	33
D	Machine Learning Methods	34
V	Details on Dataset Construction	35
A	Synthetic Networks.....	35
B	Real-world Networks	36
C	Datasets for Attack Simulations.....	36
D	Node Structural Features	37
E	Generation of True Labels and Observed Labels	39
VI	Performance metrics	40
A	Mean Absolute Error (MAE).....	40
B	Monotonicity violation frequency (M).....	41
C	Smoothness violation frequency (S).....	41
D	Additivity mean absolute error (A-MAE).....	42
VII	Model Configuration	42
A	Graph Encoder Architecture: TCR-GIN	42

B	Regressor Architecture: MLP	43
C	Training Strategy	43
VIII	Ablation Studies and Sensitivity Analysis	43
A	Structural Ablation: Multi-scale Aggregation, Residual Connections, and Virtual Node	44
B	Sensitivity of TCR-GIN to the Consistency Regularization Weight λ	45
C	Sensitivity of TCR-GIN to the Sampling Size K	46
D	Sensitivity of TCR-GIN to the Input Feature Dimension d_{in}	46
IX	Scalable Deployment Strategy on Large-Scale Networks	46
X	Lightweight Prediction of Discrete Collapse Profiles	48
XI	Supplementary Tables	50
XII	Supplementary Figures	55
	References	94

LIST OF FIGURES

S1	Verification of scale-transformed linear additivity.	55
S2	Online collapse-distance trajectories of the BA100 network under sustained attacks	56
S3	Online collapse-distance trajectories of the LFR100 network under sustained attacks	57
S4	Online collapse-distance trajectories of the ER2000 network under sustained attacks	59
S5	Online collapse-distance trajectories of the WS2000 network under sustained attacks	61
S6	Online collapse-distance trajectories of the Transport network under sustained attacks . .	63
S7	Online collapse-distance trajectories of the Power network under sustained attacks	65
S8	Sensitivity analysis of the consistency regularization weight λ	67
S9	Sensitivity analysis of the sampling size K	68
S10	Sensitivity analysis of the input feature dimension d_{in}	69
S11	Online collapse-distance trajectories of the Route network under sustained attacks	70
S12	Early-warning signal of the Transport network under 30 attack scenarios at $\tau = 0.3$	72
S13	Early-warning signal of the Power network under 30 attack scenarios at $\tau = 0.3$	75
S14	Early-warning signal of the Transport network under 30 attack scenarios at $\tau = 0.5$	78
S15	Early-warning signal of the Power network under 30 attack scenarios at $\tau = 0.5$	81
S16	Online multi-metric trajectories of the Transport network under sustained attack at $\tau = 0.3$	84
S17	Online multi-metric trajectories of the Power network under sustained attack at $\tau = 0.3$.	86
S18	Online multi-metric trajectories of the Transport network under sustained attack at $\tau = 0.5$	88
S19	Online multi-metric trajectories of the Power network under sustained attack at $\tau = 0.5$.	90
S20	Representative collapse-profile predictions on REDDIT test networks	92

LIST OF TABLES

S1	Comparison of algorithmic estimation accuracy	50
S2	Comparison of computational efficiency	51
S3	Validation against ground truth labels on small-size networks	52
S4	Ablation study results on synthetic and real-world network datasets	53
S5	Ablation study results on attack scenario datasets	54

I. MATHEMATICAL PROOFS OF COLLAPSE DISTANCE PROPERTIES

This section provides rigorous mathematical proofs for two fundamental properties of the collapse distance: monotonicity and smoothness, and scale-transformed linear additivity.

Let $G = (V, E)$ be a network with an initial total number of nodes $N_0 = |V|$. The system functionality $\Phi(G)$ is defined as the relative size of the largest connected component (LCC), i.e., $\Phi(G) = |LCC(G)|/N_0$, where $|LCC(G)|$ denotes the number of nodes in the largest connected component. For a given collapse target $\tau \in [0, 1]$, the collapse distance $D_\tau(G)$ is defined as:

$$D_\tau(G) = \min_{\mathcal{R} \subseteq V} \left\{ \frac{|\mathcal{R}|}{N_0} \mid \Phi(G \setminus \mathcal{R}) \leq \tau \right\}, \quad (1)$$

where $\mathcal{R}$ denotes the set of removed nodes and $G \setminus \mathcal{R}$ represents the remaining network after removing nodes in $\mathcal{R}$.

A. Proof of Monotonicity and Smoothness

Theorem S1 (Monotonicity and Smoothness). *The collapse distance exhibits scale-transformed monotonicity under node removals, and its evolution is smooth and predictable. Let G_t be the current network with N_t nodes, and let G_{t+1} denote the network obtained by removing an arbitrary node from G_t . To account for variations in network size, we introduce the scale correction factor $\gamma = (N_t - 1)/N_t$. Then the one-step perturbation yields a bounded change:*

$$D_\tau(G_t) - \gamma \cdot D_\tau(G_{t+1}) \in \{0, 1/N_t\}. \quad (2)$$

Proof. Let $K(G)$ denote the minimum number of nodes that must be removed to make the network G satisfy the functionality constraint $\Phi(G \setminus \mathcal{R}) \leq \tau$, i.e., the absolute number of nodes required to trigger collapse. The normalized metric is then defined as $D_\tau(G) = K(G)/|V(G)|$.

We aim to prove that $D_\tau(G_t) - \gamma D_\tau(G_{t+1}) \in \{0, 1/N_t\}$. This is equivalent to first proving that the integer-valued quantity $K(\cdot)$ satisfies $K(G_t) - K(G_{t+1}) \in \{0, 1\}$.

Step 1: Prove Non-increasing, i.e., $K(G_{t+1}) \leq K(G_t)$.

Let $\mathcal{R}_t^*$ be an optimal removal set for G_t , meaning that $|\mathcal{R}_t^*| = K(G_t)$ and $\Phi(G_t \setminus \mathcal{R}_t^*) \leq \tau$.

Consider applying this set to G_{t+1} (where $G_{t+1} = G_t \setminus \{v\}$):

$$G_{t+1} \setminus \mathcal{R}_t^* = (G_t \setminus \{v\}) \setminus \mathcal{R}_t^* = G_t \setminus (\mathcal{R}_t^* \cup \{v\}). \quad (3)$$

Due to the monotonicity of the system functionality Φ (where removing additional nodes yields a subgraph of the original configuration), we have

$$\Phi(G_t \setminus (\mathcal{R}_t^* \cup \{v\})) \leq \Phi(G_t \setminus \mathcal{R}_t^*) \leq \tau. \quad (4)$$

Consequently, the set $\mathcal{R}_t^*$ (adjusted to $\mathcal{R}_t^* \setminus \{v\}$ if $v \in \mathcal{R}_t^*$) constitutes a valid removal set for G_{t+1} . By the definition of $K(\cdot)$ as a minimum,

$$K(G_{t+1}) \leq |\mathcal{R}_t^*| = K(G_t). \quad (5)$$

Step 2: Prove the bounded variation, i.e., $K(G_t) \leq K(G_{t+1}) + 1$.

Let $\mathcal{R}_{t+1}^*$ be an optimal removal set for G_{t+1} , meaning that $|\mathcal{R}_{t+1}^*| = K(G_{t+1})$ and $\Phi(G_{t+1} \setminus \mathcal{R}_{t+1}^*) \leq \tau$.

We construct a candidate removal set for G_t based on $\mathcal{R}_{t+1}^*$. Construct the set $\mathcal{R}' = \mathcal{R}_{t+1}^* \cup \{v\}$. Applying $\mathcal{R}'$ to G_t :

$$G_t \setminus \mathcal{R}' = G_t \setminus (\mathcal{R}_{t+1}^* \cup \{v\}) = (G_t \setminus \{v\}) \setminus \mathcal{R}_{t+1}^* = G_{t+1} \setminus \mathcal{R}_{t+1}^*. \quad (6)$$

Since $\mathcal{R}_{t+1}^*$ is a valid solution for G_{t+1} , we have

$$\Phi(G_t \setminus \mathcal{R}') = \Phi(G_{t+1} \setminus \mathcal{R}_{t+1}^*) \leq \tau. \quad (7)$$

This shows that $\mathcal{R}'$ is a valid removal set for G_t . By the definition of minimality, the optimal solution for G_t , $K(G_t)$, must be less than or equal to the size of $\mathcal{R}'$:

$$K(G_t) \leq |\mathcal{R}'| = |\mathcal{R}_{t+1}^*| + 1 = K(G_{t+1}) + 1. \quad (8)$$

Rearranging terms yields the inequality,

$$K(G_t) - K(G_{t+1}) \leq 1. \quad (9)$$

Step 3: Introduce the scale-correction factor.

Combining the results of Step 1 and Step 2, we obtain the integer constraint $K(G_t) - K(G_{t+1}) \in \{0, 1\}$.

Now consider the difference term in the theorem. Substituting $D_\tau(G) = K(G)/|V(G)|$, we have

$$\begin{aligned} \Delta &= D_\tau(G_t) - \gamma \cdot D_\tau(G_{t+1}) \\ &= \frac{K(G_t)}{N_t} - \left(\frac{N_t - 1}{N_t} \right) \cdot \frac{K(G_{t+1})}{N_t - 1} \\ &= \frac{K(G_t)}{N_t} - \frac{K(G_{t+1})}{N_t} \\ &= \frac{1}{N_t} (K(G_t) - K(G_{t+1})). \end{aligned} \quad (10)$$

Given that $K(G)$ is integer-valued and we have established $K(G_t) - K(G_{t+1}) \in \{0, 1\}$, it follows that

$$\Delta \in \{0, 1/N_t\}. \quad (11)$$

This completes the proof. $\square$

B. Proof of Scale-Transformed Linear Additivity

Theorem S2 (Scale-Transformed Linear Additivity). *Suppose a network $G = (V, E)$ can be decomposed into the disjoint union of k connected components, $G = \bigsqcup_{j=1}^k C_j$. Let $n_j := |V(C_j)|$ denote the number of nodes in component C_j , with $\sum_{j=1}^k n_j = N$. Then the collapse distance $D_\tau(G)$ satisfies:*

$$D_\tau(G) = \sum_{j=1}^k \frac{n_j}{N} D_{\tau \cdot \frac{N}{n_j}}(C_j), \quad (12)$$

where we adopt the convention that $D_{\tau'}(\cdot) = 0$ whenever $\tau' \geq 1$, since in this regime the constraint is automatically satisfied without any node removal.

Proof. The proof rests on decomposing the global constrained optimization problem that defines $D_\tau(G)$ into a set of independent local optimization problems for each connected component.

By definition, the global collapse distance is the solution to the following constrained optimization problem:

$$\min_{\mathcal{R} \subseteq V} \frac{|\mathcal{R}|}{N} \quad \text{subject to} \quad \frac{|LCC(G \setminus \mathcal{R})|}{N} \leq \tau. \quad (13)$$

Note that this problem is always feasible: in the worst case, removing all nodes (i.e., $\mathcal{R} = V$) yields $|LCC(G \setminus V)| = 0 \leq \tau N$.

The constraint is equivalent to imposing an upper bound on the absolute size of the largest connected component:

$$|LCC(G \setminus \mathcal{R})| \leq \tau N. \quad (14)$$

Suppose the network G consists of k topologically disjoint connected components, denoted by $C_1, \dots, C_k$, and let $n_j = |V(C_j)|$ and $N = \sum_{j=1}^k n_j$. Since these components are topologically isolated, any removal set $\mathcal{R}$ uniquely decomposes into $\mathcal{R} = \bigsqcup_{j=1}^k \mathcal{R}_j$, where $\mathcal{R}_j = \mathcal{R} \cap V(C_j)$. Consequently, the largest connected component of the remaining network $G \setminus \mathcal{R}$ must be the largest remaining connected component of one of the subgraphs $C_j \setminus \mathcal{R}_j$, hence:

$$|LCC(G \setminus \mathcal{R})| = \max_{j=1, \dots, k} \{|LCC(C_j \setminus \mathcal{R}_j)|\}. \quad (15)$$

Therefore, the global constraint $\max_j \{|LCC(C_j \setminus \mathcal{R}_j)|\} \leq \tau N$ is equivalent to requiring every component to independently satisfy the same absolute size threshold:

$$|LCC(C_j \setminus \mathcal{R}_j)| \leq \tau N, \quad \forall j \in \{1, \dots, k\}. \quad (16)$$

Simultaneously, the objective function (fraction of removed nodes) can be rewritten as a node-weighted sum of the fractions removed from each component:

$$\frac{|\mathcal{R}|}{N} = \sum_{j=1}^k \frac{|\mathcal{R}_j|}{N} = \sum_{j=1}^k \frac{n_j}{N} \left(\frac{|\mathcal{R}_j|}{n_j} \right). \quad (17)$$

At this stage, both the objective function and the constraints are fully separable across the components: the objective is a sum of terms each depending only on $\mathcal{R}_j$, and the constraint for each component involves only $\mathcal{R}_j$. By the separability principle for constrained optimization, the global optimum equals the sum of the component-wise optima:

$$\min_{\mathcal{R}_1, \dots, \mathcal{R}_k} \sum_{j=1}^k \frac{n_j}{N} \left(\frac{|\mathcal{R}_j|}{n_j} \right) = \sum_{j=1}^k \frac{n_j}{N} \min_{\mathcal{R}_j \subseteq V(C_j)} \frac{|\mathcal{R}_j|}{n_j}, \quad (18)$$

where each minimization is subject to $|LCC(C_j \setminus \mathcal{R}_j)| \leq \tau N$.

It remains to express each component-wise subproblem in the standard form of the collapse distance. Recall that for a connected network H with $|V(H)| = n$ and a threshold parameter τ' , the collapse distance is defined as:

$$D_{\tau'}(H) := \min_{\mathcal{R} \subseteq V(H)} \frac{|\mathcal{R}|}{n} \quad \text{subject to} \quad \frac{|LCC(H \setminus \mathcal{R})|}{n} \leq \tau', \quad (19)$$

or equivalently, subject to $|LCC(H \setminus \mathcal{R})| \leq \tau' n$. To align the absolute constraint $|LCC(C_j \setminus \mathcal{R}_j)| \leq \tau N$ with this standard form $|LCC(C_j \setminus \mathcal{R}_j)| \leq \tau' n_j$, we set $\tau' n_j = \tau N$, yielding the effective threshold $\tau' = \tau \cdot N/n_j$.

Note that when $n_j \leq \tau N$, the effective threshold satisfies $\tau' \geq 1$, in which case the constraint is automatically satisfied for any $\mathcal{R}_j$ (since $|LCC(C_j \setminus \mathcal{R}_j)| \leq n_j \leq \tau N$), so $D_{\tau \cdot N/n_j}(C_j) = 0$ and no nodes need to be removed from component C_j .

Consequently, the optimal removal fraction for each component C_j corresponds exactly to its collapse distance under this effective relative threshold, i.e., $D_{\tau \cdot (N/n_j)}(C_j)$.

Substituting these component-wise optima back into the global objective gives,

$$D_\tau(G) = \sum_{j=1}^k \frac{n_j}{N} D_{\tau \cdot \frac{N}{n_j}}(C_j). \quad (20)$$

This completes the proof. □

II. THEORETICAL ANALYSIS OF TCR-GIN FRAMEWORK

This section develops the theoretical underpinnings of the TCR-GIN estimator from five angles, organized to mirror the four questions posed in the main text. Subsection II A proves a sufficient condition under which a learner trained on one-sided biased observed labels incurs strictly smaller true-label mean-absolute error (MAE) than every individual supervision heuristic. Subsection II B derives explicit upper bounds on both the true-label MAE and the true-label mean-squared-error (MSE) risk of TCR-GIN, and shows that the supervision-bias components of these bounds contract monotonically as new dismantling heuristics are incorporated into the supervision pool. Subsection II C isolates the complementary role of topological-consistency regularization. Analyzed at the population level as a convex bounded-variation surrogate, the regularizer reduces a component of the supervision bias that persists in the infinite-sample limit under fixed observed labels and therefore cannot be removed by increasing the training-set size alone. Subsection II D bounds the additivity error of the learned prediction under network fragmentation, connecting the nonlinear estimator to the scale-transformed linear additivity of CD (Theorem S2). Subsection II E reports the forward-pass computational complexity of TCR-GIN.

A. Sufficient Condition for Surpassing Individual Heuristics under One-Sided Label Bias

In this subsection, we prove a sufficient condition under which the learned predictor incurs strictly smaller true-label MAE than every individual heuristic, despite never seeing true labels during training.

Let $G \sim \mathcal{D}$ be a random graph and let $D_\tau(G) \in [0, 1]$ denote the collapse distance, i.e., the true label. Let $\mathcal{H} = \{h_1, \dots, h_m\}$ be a finite set of dismantling heuristics. Each heuristic provides an upper-bound estimate

$$\hat{D}^h(G) \geq D_\tau(G), \quad h \in \mathcal{H}. \quad (21)$$

The observed label used for supervision is the pointwise minimum over the available heuristic estimates:

$$\tilde{D}_\tau(G; \mathcal{H}) := \min_{h \in \mathcal{H}} \hat{D}^h(G). \quad (22)$$

The supervision bias is

$$\eta(G; \mathcal{H}) := \tilde{D}_\tau(G; \mathcal{H}) - D_\tau(G) \geq 0. \quad (23)$$

We use MAE as the final evaluation metric. The observed-label MAE measures how well a predictor fits the heuristic-derived training label, whereas the true-label MAE measures the final

prediction error with respect to the exact collapse distance:

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) := \mathbb{E}|f(G) - \tilde{D}_\tau(G; \mathcal{H})|, \quad \mathcal{E}_{\text{true}}(f) := \mathbb{E}|f(G) - D_\tau(G)|. \quad (24)$$

The training objective is based on squared loss with respect to the observed label. Accordingly, the observed-label MSE is the population counterpart of the training loss, while the true-label MSE is introduced only as an analytical reference for squared error with respect to the exact collapse distance:

$$\mathcal{K}_{\text{obs}}(f; \mathcal{H}) := \mathbb{E}[(f(G) - \tilde{D}_\tau(G; \mathcal{H}))^2], \quad \mathcal{K}_{\text{true}}(f) := \mathbb{E}[(f(G) - D_\tau(G))^2]. \quad (25)$$

Thus $\mathcal{K}_{\text{obs}}$ is the risk directly optimized by observed-label training, whereas $\mathcal{E}_{\text{true}}$ is the MAE metric used to evaluate whether the learned predictor improves over individual heuristics on the true collapse distance. Because $f(G)$, $D_\tau(G)$, and $\tilde{D}_\tau(G; \mathcal{H})$ lie in $[0, 1]$, Jensen's inequality gives

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) \leq \sqrt{\mathcal{K}_{\text{obs}}(f; \mathcal{H})}, \quad \mathcal{E}_{\text{true}}(f) \leq \sqrt{\mathcal{K}_{\text{true}}(f)}. \quad (26)$$

Therefore, minimizing the observed-label MSE controls the observed-label MAE. But it does not guarantee by itself improvement in true-label MAE. The role of Theorem S3 is precisely to characterize when this improvement transfers from observed-label fitting to true-label evaluation under one-sided supervision bias.

When a heuristic $h \in \mathcal{H}$ is viewed as the predictor $G \mapsto \hat{D}^h(G)$, we write

$$\mathcal{E}_{\text{obs}}(h; \mathcal{H}) := \mathcal{E}_{\text{obs}}(\hat{D}^h; \mathcal{H}), \quad \mathcal{E}_{\text{true}}(h) := \mathcal{E}_{\text{true}}(\hat{D}^h). \quad (27)$$

Theorem S3 (True-label MAE improvement over individual heuristics). *For any measurable predictor f , define its one-sided bias slack by*

$$\Gamma(f; \mathcal{H}) := \mathbb{E} \left[\eta(G; \mathcal{H}) - \left(|f(G) - D_\tau(G)| - |f(G) - \tilde{D}_\tau(G; \mathcal{H})| \right) \right]. \quad (28)$$

If

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) < \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}) + \Gamma(f; \mathcal{H}), \quad (29)$$

then

$$\mathcal{E}_{\text{true}}(f) < \min_{h \in \mathcal{H}} \mathcal{E}_{\text{true}}(h). \quad (30)$$

Proof. For every graph G , the definitions of $\tilde{D}_\tau(G; \mathcal{H})$ and $\eta(G; \mathcal{H})$ give

$$f(G) - D_\tau(G) = (f(G) - \tilde{D}_\tau(G; \mathcal{H})) + \eta(G; \mathcal{H}). \quad (31)$$

Applying the triangle inequality to Equation 31 yields

$$|f(G) - D_\tau(G)| \leq |f(G) - \tilde{D}_\tau(G; \mathcal{H})| + \eta(G; \mathcal{H}). \quad (32)$$

Equivalently,

$$|f(G) - D_\tau(G)| - |f(G) - \tilde{D}_\tau(G; \mathcal{H})| \leq \eta(G; \mathcal{H}). \quad (33)$$

Equation 33 shows that the integrand in Equation 28 is non-negative. Hence

$$\Gamma(f; \mathcal{H}) \geq 0. \quad (34)$$

Rearranging Equation 28 and using the definitions in Equation 24 gives the exact identity

$$\mathcal{E}_{\text{true}}(f) = \mathcal{E}_{\text{obs}}(f; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})] - \Gamma(f; \mathcal{H}). \quad (35)$$

Next, fix any heuristic $h \in \mathcal{H}$. By the upper-bound property in Equation 21 and the observed-label definition in Equation 22, we have

$$\hat{D}^h(G) \geq \tilde{D}_\tau(G; \mathcal{H}) \geq D_\tau(G). \quad (36)$$

Therefore the absolute values have fixed signs:

$$|\hat{D}^h(G) - D_\tau(G)| = \hat{D}^h(G) - D_\tau(G), \quad |\hat{D}^h(G) - \tilde{D}_\tau(G; \mathcal{H})| = \hat{D}^h(G) - \tilde{D}_\tau(G; \mathcal{H}). \quad (37)$$

Subtracting the two equalities in Equation 37 gives

$$|\hat{D}^h(G) - D_\tau(G)| - |\hat{D}^h(G) - \tilde{D}_\tau(G; \mathcal{H})| = \tilde{D}_\tau(G; \mathcal{H}) - D_\tau(G) = \eta(G; \mathcal{H}), \quad (38)$$

where the last equality uses Equation 23. Taking expectations in Equation 38 and using the shorthand in Equation 27 yields

$$\mathcal{E}_{\text{true}}(h) = \mathcal{E}_{\text{obs}}(h; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})]. \quad (39)$$

Because the additive term $\mathbb{E}[\eta(G; \mathcal{H})]$ in Equation 39 is the same for every $h \in \mathcal{H}$, taking the minimum over h gives

$$\min_{h \in \mathcal{H}} \mathcal{E}_{\text{true}}(h) = \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})]. \quad (40)$$

Now suppose Equation 29 holds. Combining Equation 35 with Equation 29 gives

$$\begin{aligned} \mathcal{E}_{\text{true}}(f) &= \mathcal{E}_{\text{obs}}(f; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})] - \Gamma(f; \mathcal{H}) \\ &< \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}) + \Gamma(f; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})] - \Gamma(f; \mathcal{H}) \\ &= \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})]. \end{aligned} \quad (41)$$

Finally, substituting Equation 40 into Equation 41 proves Equation 30. $\square$

Remark S1 (Useful sufficient conditions). The condition in Equation 29 has two useful special cases. First, since Equation 34 shows $\Gamma(f; \mathcal{H}) \geq 0$, strict improvement follows if

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) < \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}). \quad (42)$$

Second, strict improvement also follows under the weaker observed-label condition

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) \leq \min_{h \in \mathcal{H}} \mathcal{E}_{\text{obs}}(h; \mathcal{H}) \quad \text{and} \quad \Gamma(f; \mathcal{H}) > 0. \quad (43)$$

The second case is the one that most directly reflects the one-sided nature of the supervision. The learned predictor can tie the best heuristic on observed-label MAE and still improve on true-label MAE if it inherits strictly less upward bias.

Remark S2 (Interpretation). Theorem S3 gives a sufficient mechanism, not an automatic guarantee, for surpassing every individual heuristic. Individual heuristics always lie above the observed lower envelope by Equation 36. Therefore, they inherit the full bias gap $\mathbb{E}[\eta(G; \mathcal{H})]$, as shown by Equation 39. A learned predictor is not constrained to remain above the observed label, so it may have positive slack in Equation 28 and inherit less upward bias. Consequently, the learner can beat every individual heuristic even when its observed-label MAE only ties the best heuristic, provided that the one-sided slack is positive. The smooth inductive bias of neural-network learning provides a natural mechanism for such positive slack.

B. Bounded True-Label Error and Monotone Contraction under Heuristic-Pool Expansion

The sufficient condition of Theorem S3 is stated for a fixed heuristic set $\mathcal{H}$. Two natural questions arise: (i) does the true-label prediction error admit an explicit, computable upper bound? and (ii) does this bound improve as stronger heuristics become available? We answer both affirmatively. Theorem S4 first shows that the supervision gap tightens pointwise as $\mathcal{H}$ grows. Theorem S5 then derives an explicit true-label MAE upper bound whose supervision-bias components are monotone non-increasing along any expanding sequence of heuristic sets, so the theoretical guarantee co-evolves with advances in dismantling algorithms. And that is also the case for the true-label MSE (Corollary S1).

We adopt throughout this subsection the notation introduced in Section II A (D_τ , $\mathcal{H}$, $\tilde{D}_\tau(\cdot; \mathcal{H})$, $\eta(\cdot; \mathcal{H})$, $\mathcal{D}$), and additionally introduce the following objects. Let $\mathcal{H} = \{h_1, h_2, \dots\}$ denote a countable family of admissible dismantling heuristics, of which each finite $\mathcal{H}$ used in practice is a

subset. A sequence $\{\mathcal{H}_k\}_{k \geq 1}$ is monotonically expanding if $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots \subset \mathcal{H}$. Let $\mathcal{F}$ be a uniformly bounded hypothesis class of measurable functions $f : G \mapsto f(G) \in [0, 1]$ representing the family of admissible TCR-GIN predictors. In practice, exact minimization over $\mathcal{F}$ may not be achieved. We say that $\hat{f}_n^{\mathcal{H}}$ is an $\varepsilon_{\text{opt}}(\mathcal{H})$ -approximate ideal empirical-risk minimizer if

$$\hat{\mathcal{K}}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) \leq \inf_{f \in \mathcal{F}} \hat{\mathcal{K}}_{\text{obs}}(f; \mathcal{H}) + \varepsilon_{\text{opt}}(\mathcal{H}), \quad (44)$$

where $\varepsilon_{\text{opt}}(\mathcal{H}) \geq 0$ is the optimization tolerance and

$$\hat{\mathcal{K}}_{\text{obs}}(f; \mathcal{H}) := \frac{1}{n} \sum_{i=1}^n (f(G_i) - \tilde{D}_{\tau}(G_i; \mathcal{H}))^2 \quad (45)$$

is the empirical observed-label MSE. Given n i.i.d. training graphs $\{G_i\}_{i=1}^n \sim \mathcal{D}^{\otimes n}$, the empirical risk minimization (ERM) under supervision $\mathcal{H}$ is

$$\hat{f}_n^{\mathcal{H}} \in \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (f(G_i) - \tilde{D}_{\tau}(G_i; \mathcal{H}))^2, \quad (46)$$

which corresponds exactly to the MSE loss used during TCR-GIN training. The corresponding population-level true-label MSE and MAE are $\mathcal{K}_{\text{true}}(f) := \mathbb{E}_{G \sim \mathcal{D}}[(f(G) - D_{\tau}(G))^2]$ and $\mathcal{E}_{\text{true}}(f) := \mathbb{E}_{G \sim \mathcal{D}}|f(G) - D_{\tau}(G)|$, respectively. Since the main text reports MAE as the evaluation metric, Theorem S5 is stated directly at the MAE level, and the MSE guarantee follows as a corollary.

Theorem S4 (Observed-label gap tightens monotonically under heuristic-set expansion). *For any heuristic sets $\mathcal{H} \subseteq \mathcal{H}' \subset \mathcal{H}$, the pointwise supervision gap satisfies, for $\mathcal{D}$ -almost every G ,*

$$0 \leq \eta(G; \mathcal{H}') \leq \eta(G; \mathcal{H}). \quad (47)$$

Proof. Fix $G \in \text{supp}(\mathcal{D})$. Since $\mathcal{H} \subseteq \mathcal{H}'$, the minimization defining $\tilde{D}_{\tau}(G; \mathcal{H}')$ ranges over a superset of the candidates used in $\tilde{D}_{\tau}(G; \mathcal{H})$. Enlarging the candidate set of a pointwise minimum can only decrease its value, hence

$$\tilde{D}_{\tau}(G; \mathcal{H}') = \min_{h \in \mathcal{H}'} \hat{D}^h(G) \leq \min_{h \in \mathcal{H}} \hat{D}^h(G) = \tilde{D}_{\tau}(G; \mathcal{H}). \quad (48)$$

The upper-bound property $\hat{D}^h(G) \geq D_{\tau}(G)$ holds for every admissible h , and the minimum of a family of upper bounds remains an upper bound, so $\tilde{D}_{\tau}(G; \mathcal{H}') \geq D_{\tau}(G)$. Subtracting $D_{\tau}(G)$ from both sides of Equation 48 yields Equation 47. $\square$

The map $x \mapsto x^p$ is monotone non-decreasing on $[0, \infty)$ for every $p \geq 1$. Applying it to Equation 47 and integrating against $\mathcal{D}$ gives,

$$\mathbb{E}_{G \sim \mathcal{D}}[\eta(G; \mathcal{H}')^p] \leq \mathbb{E}_{G \sim \mathcal{D}}[\eta(G; \mathcal{H})^p], \quad \forall p \geq 1 \quad \text{and} \quad \mathcal{H} \subseteq \mathcal{H}' \subset \mathcal{H}. \quad (49)$$

Theorem S4 captures an elementary but foundational fact that enlarging $\mathcal{H}$ can never make the supervision worse, since the observed label can only move closer to the true CD and never further away. The next theorem propagates this pointwise tightening to the learner’s true-label MAE.

We next convert the monotone contraction of $\eta(G; \mathcal{H})$ into a true-label MAE guarantee. Let

$$\mathfrak{R}_n(\mathcal{F}) := \mathbb{E}_{\{G_i\}, \{\sigma_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(G_i) \right], \quad (50)$$

where σ_i are independent Rademacher signs. For bounded GIN architectures with finite pseudo-dimension or finite covering numbers, $\mathfrak{R}_n(\mathcal{F}) = O(n^{-1/2})$ up to logarithmic factors.

Theorem S5 (True-label MAE is bounded and contracts with heuristic-pool expansion). *Assume $\mathcal{F}$ is uniformly bounded by 1. Fix $n \geq 1$ and $\delta \in (0, 1)$. For every fixed finite $\mathcal{H} \subset \mathcal{H}$, with probability at least $1 - \delta$ over the training sample, any predictor satisfying Equation 44 obeys*

$$\mathcal{E}_{\text{true}}(\hat{f}_n^{\mathcal{H}}) \leq \mathcal{M}_{n,\delta}(\mathcal{H}) := \sqrt{2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F}) + \sqrt{\frac{2\log(2/\delta)}{n}} + 2\mathbb{E}[\eta(G; \mathcal{H})^2] + \varepsilon_{\text{opt}}(\mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})]}, \quad (51)$$

where

$$\mathcal{A}^* := \inf_{f \in \mathcal{F}} \mathcal{K}_{\text{true}}(f) = \inf_{f \in \mathcal{F}} \mathbb{E}[(f(G) - D_\tau(G))^2]. \quad (52)$$

By Equation 51, apart from the optimization tolerance, the true-label MAE guarantee depends on the expanding heuristic set through the second-moment supervision bias $\mathbb{E}[\eta(G; \mathcal{H})^2]$ inside the square root and the mean supervision bias $\mathbb{E}[\eta(G; \mathcal{H})]$ as an additive term. By Theorem S4 and Equation 49, both quantities are monotone non-increasing along any monotonically expanding sequence $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$. Therefore, the supervision-bias components of the true-label MAE guarantee contract monotonically as the heuristic pool expands. The full upper bound is monotone non-increasing if the optimization tolerance is held fixed, is non-increasing, or increases slowly enough to be dominated by the contraction of the supervision-bias terms. More precisely, writing

$$b_k := \mathbb{E}[\eta(G; \mathcal{H}_k)^2], \quad m_k := \mathbb{E}[\eta(G; \mathcal{H}_k)], \quad \varepsilon_k := \varepsilon_{\text{opt}}(\mathcal{H}_k), \quad (53)$$

and

$$C_0 := 2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F}) + \sqrt{\frac{2\log(2/\delta)}{n}}, \quad (54)$$

the bound reads $\mathcal{M}_{n,\delta}(\mathcal{H}_k) = \sqrt{C_0 + 2b_k + \varepsilon_k} + m_k$ and satisfies

$$\mathcal{M}_{n,\delta}(\mathcal{H}_{k+1}) \leq \mathcal{M}_{n,\delta}(\mathcal{H}_k) \quad \text{whenever} \quad \varepsilon_{k+1} - \varepsilon_k \leq 2(b_k - b_{k+1}). \quad (55)$$

If $\varepsilon_{\text{opt}}(\mathcal{H}_k) \leq \bar{\varepsilon}_{\text{opt}}$ for all k , then the common-tolerance bound obtained by replacing $\varepsilon_{\text{opt}}(\mathcal{H}_k)$ with $\bar{\varepsilon}_{\text{opt}}$ is monotone non-increasing. If, in addition, the expanding family contains heuristic estimates converging to D_τ in $L^2(\mathcal{D})$, then the common-tolerance bound converges to

$$\sqrt{2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F})} + \sqrt{\frac{2\log(2/\delta)}{n}} + \bar{\varepsilon}_{\text{opt}}. \quad (56)$$

Proof. Fix a finite heuristic set $\mathcal{H}$. The proof decomposes the true-label MAE into an observed-label MAE term and a mean supervision-bias term, bridges the observed-label MAE to the observed-label MSE via Jensen's inequality, controls the observed-label MSE by a Rademacher uniform-deviation argument, and replaces the heuristic-dependent approximation term by an $\mathcal{H}$ -free quantity to expose the monotonicity.

Step 1. Triangle-inequality decomposition from true label to observed label. For any $f \in \mathcal{F}$, using $\tilde{D}_\tau(G; \mathcal{H}) - D_\tau(G) = \eta(G; \mathcal{H})$ gives

$$f(G) - D_\tau(G) = (f(G) - \tilde{D}_\tau(G; \mathcal{H})) + \eta(G; \mathcal{H}). \quad (57)$$

Applying the triangle inequality $|a + b| \leq |a| + |b|$ to Equation 57 and taking expectations yields

$$\mathcal{E}_{\text{true}}(f) \leq \mathcal{E}_{\text{obs}}(f; \mathcal{H}) + \mathbb{E}[\eta(G; \mathcal{H})], \quad (58)$$

which we will apply at $f = \hat{f}_n^\mathcal{H}$.

Step 2. Jensen bridge from observed-label MAE to observed-label MSE. Since $f(G)$ and $\tilde{D}_\tau(G; \mathcal{H})$ both lie in $[0, 1]$, the residual $|f(G) - \tilde{D}_\tau(G; \mathcal{H})|$ takes values in $[0, 1]$. Jensen's inequality applied to the concave function $\sqrt{\cdot}$ gives

$$\mathcal{E}_{\text{obs}}(f; \mathcal{H}) = \mathbb{E}|f(G) - \tilde{D}_\tau(G; \mathcal{H})| \leq \sqrt{\mathbb{E}[(f(G) - \tilde{D}_\tau(G; \mathcal{H}))^2]} = \sqrt{\mathcal{K}_{\text{obs}}(f; \mathcal{H})}. \quad (59)$$

Substituting into Equation 58 yields

$$\mathcal{E}_{\text{true}}(f) \leq \sqrt{\mathcal{K}_{\text{obs}}(f; \mathcal{H})} + \mathbb{E}[\eta(G; \mathcal{H})]. \quad (60)$$

It therefore suffices to bound $\mathcal{K}_{\text{obs}}(\hat{f}_n^\mathcal{H}; \mathcal{H})$, the population observed-label MSE of the approximate ERM, which is the population counterpart of the training objective.

Step 3. Uniform deviation via McDiarmid, symmetrization, and contraction. We control the population observed-label MSE by its empirical counterpart. Define the loss class $\ell_f(G) := (f(G) - \tilde{D}_\tau(G; \mathcal{H}))^2$. Since both $f(G)$ and $\tilde{D}_\tau(G; \mathcal{H})$ lie in $[0, 1]$, we have $\ell_f(G) \in [0, 1]$, and the

map $u \mapsto (u - \tilde{D}_\tau(G; \mathcal{H}))^2$ is 2-Lipschitz on $[0, 1]$. A standard bounded-difference, symmetrization, and contraction argument therefore gives, with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} \left| \mathcal{K}_{\text{obs}}(f; \mathcal{H}) - \hat{\mathcal{K}}_{\text{obs}}(f; \mathcal{H}) \right| \leq \Delta_{n,\delta} := 4\mathfrak{R}_n(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2n}}. \quad (61)$$

Step 4. Approximate ERM optimality. On the event in Equation 61, the approximate empirical minimizer satisfies

$$\begin{aligned} \mathcal{K}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) &\leq \hat{\mathcal{K}}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) + \Delta_{n,\delta} \\ &\leq \inf_{f \in \mathcal{F}} \hat{\mathcal{K}}_{\text{obs}}(f; \mathcal{H}) + \varepsilon_{\text{opt}}(\mathcal{H}) + \Delta_{n,\delta} \\ &\leq \inf_{f \in \mathcal{F}} \mathcal{K}_{\text{obs}}(f; \mathcal{H}) + \varepsilon_{\text{opt}}(\mathcal{H}) + 2\Delta_{n,\delta}. \end{aligned} \quad (62)$$

Let

$$\mathcal{A}_{\mathcal{H}} := \inf_{f \in \mathcal{F}} \mathcal{K}_{\text{obs}}(f; \mathcal{H}). \quad (63)$$

Then Equation 62 can be written as

$$\mathcal{K}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) \leq \mathcal{A}_{\mathcal{H}} + \varepsilon_{\text{opt}}(\mathcal{H}) + 2\Delta_{n,\delta}. \quad (64)$$

Step 5. Replacing $\mathcal{A}_{\mathcal{H}}$ by a manifestly $\mathcal{H}$ -monotone bound. The observed-label oracle risk $\mathcal{A}_{\mathcal{H}}$ need not itself be monotone in $\mathcal{H}$, so we upper-bound it by a monotone expression. For any $f \in \mathcal{F}$,

$$f(G) - \tilde{D}_\tau(G; \mathcal{H}) = (f(G) - D_\tau(G)) - \eta(G; \mathcal{H}). \quad (65)$$

Using $(a - b)^2 \leq 2a^2 + 2b^2$ in Equation 65, taking expectations, and then taking the infimum over $f \in \mathcal{F}$ gives

$$\mathcal{A}_{\mathcal{H}} \leq 2\mathcal{A}^* + 2\mathbb{E}[\eta(G; \mathcal{H})^2]. \quad (66)$$

Step 6. Assembly. Substituting Equation 66 into Equation 64 gives the explicit observed-label MSE bound

$$\mathcal{K}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) \leq 2\mathcal{A}^* + \varepsilon_{\text{opt}}(\mathcal{H}) + 2\Delta_{n,\delta} + 2\mathbb{E}[\eta(G; \mathcal{H})^2]. \quad (67)$$

Substituting the definition of $\Delta_{n,\delta}$ from Equation 61, applying the Jensen bridge (Equation 59) to $\hat{f}_n^{\mathcal{H}}$ via

$$\mathcal{E}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H}) \leq \sqrt{\mathcal{K}_{\text{obs}}(\hat{f}_n^{\mathcal{H}}; \mathcal{H})} \leq \sqrt{2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F}) + \sqrt{\frac{2\log(2/\delta)}{n}} + 2\mathbb{E}[\eta(G; \mathcal{H})^2] + \varepsilon_{\text{opt}}(\mathcal{H})}, \quad (68)$$

and substituting into the triangle-inequality decomposition (Equation 58) yields

$$\mathcal{E}_{\text{true}}(\hat{f}_n^{\mathcal{H}}) \leq \underbrace{\sqrt{2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F})} + \sqrt{\frac{2\log(2/\delta)}{n}} + 2\mathbb{E}[\eta(G; \mathcal{H})^2] + \varepsilon_{\text{opt}}(\mathcal{H})}_{\text{observed-label fitting envelope}} + \underbrace{\mathbb{E}[\eta(G; \mathcal{H})]}_{\text{mean supervision bias}}, \quad (69)$$

which is the claimed bound $\mathcal{M}_{n,\delta}(\mathcal{H})$ of Equation 51.

Step 7. Monotonicity and convergence. In Equation 69, the two $\mathcal{H}$ -dependent quantities (apart from $\varepsilon_{\text{opt}}(\mathcal{H})$) are $\mathbb{E}[\eta(\cdot; \mathcal{H})^2]$ and $\mathbb{E}[\eta(\cdot; \mathcal{H})]$. By Theorem S4 and Equation 49 with $p = 2$ and $p = 1$ respectively, both quantities are monotone non-increasing along any monotonically expanding sequence:

$$b_{k+1} \leq b_k, \quad m_{k+1} \leq m_k. \quad (70)$$

Since $\sqrt{\cdot}$ is monotone non-decreasing on $[0, \infty)$, the square-root term $\sqrt{C_0 + 2b_k + \varepsilon_k}$ is non-increasing whenever $2b_{k+1} + \varepsilon_{k+1} \leq 2b_k + \varepsilon_k$, i.e., $\varepsilon_{k+1} - \varepsilon_k \leq 2(b_k - b_{k+1})$. Because $m_{k+1} \leq m_k$, the additive mean-bias term is also non-increasing. Hence the full bound $\mathcal{M}_{n,\delta}(\mathcal{H}_{k+1}) \leq \mathcal{M}_{n,\delta}(\mathcal{H}_k)$ precisely when Equation 55 holds. In particular, the full bound is non-increasing if $\varepsilon_{\text{opt}}(\mathcal{H}_k)$ is held fixed or is non-increasing.

If only a common optimization-tolerance upper bound $\varepsilon_{\text{opt}}(\mathcal{H}_k) \leq \bar{\varepsilon}_{\text{opt}}$ is available, replacing $\varepsilon_{\text{opt}}(\mathcal{H}_k)$ by $\bar{\varepsilon}_{\text{opt}}$ gives a common-tolerance bound whose $\mathcal{H}_k$ -dependent terms are $2b_k$ (inside the square root) and m_k (additive). Both are non-increasing by Equation 70, so the common-tolerance bound is monotone non-increasing. For the convergence claim, suppose $\{h_j\} \subset \bigcup_k \mathcal{H}_k$ satisfies $\hat{D}^{h_j} \rightarrow D_\tau$ in $L^2(\mathcal{D})$. For each k pick $j(k)$ with $h_{j(k)} \in \mathcal{H}_k$ along the convergent subsequence. The pointwise squeeze $D_\tau(G) \leq \tilde{D}_\tau(G; \mathcal{H}_k) \leq \hat{D}^{h_{j(k)}}(G)$ gives $|\tilde{D}_\tau(G; \mathcal{H}_k) - D_\tau(G)| \leq |\hat{D}^{h_{j(k)}}(G) - D_\tau(G)|$; taking $L^2(\mathcal{D})$ -norms yields $\mathbb{E}[\eta(\cdot; \mathcal{H}_k)^2] \rightarrow 0$. By Jensen's inequality, $\mathbb{E}[\eta(\cdot; \mathcal{H}_k)] \leq \sqrt{\mathbb{E}[\eta(\cdot; \mathcal{H}_k)^2]} \rightarrow 0$, so the common-tolerance bound converges to $\sqrt{2\mathcal{A}^* + 8\mathfrak{R}_n(\mathcal{F})} + \sqrt{2\log(2/\delta)/n} + \bar{\varepsilon}_{\text{opt}}$. $\square$

Corollary S1 (True-label MSE is bounded and inherits monotone contraction). *Under the hypotheses of Theorem S5, with the same probability at least $1 - \delta$, the true-label MSE risk of $\hat{f}_n^{\mathcal{H}}$ satisfies*

$$\mathcal{K}_{\text{true}}(\hat{f}_n^{\mathcal{H}}) \leq \mathcal{M}_{n,\delta}(\mathcal{H}). \quad (71)$$

In particular, the MSE bound inherits the same monotonicity and convergence properties as the MAE bound.

Proof. Since $f(G)$ and $D_\tau(G)$ both lie in $[0, 1]$, the residual $|f(G) - D_\tau(G)|$ takes values in $[0, 1]$. The monotonicity $t^2 \leq t$ for $t \in [0, 1]$ applied pointwise and integrated against $\mathcal{D}$ gives

$$\mathcal{K}_{\text{true}}(f) = \mathbb{E}[(f(G) - D_\tau(G))^2] \leq \mathbb{E}|f(G) - D_\tau(G)| = \mathcal{E}_{\text{true}}(f). \quad (72)$$

Applying Equation 72 at $f = \hat{f}_n^{\mathcal{H}}$ and invoking Theorem S5 yields Equation 71. The monotonicity and convergence claims follow immediately from those of $\mathcal{M}_{n,\delta}(\mathcal{H})$ established in Theorem S5. $\square$

Remark S3 (Fixed-set and simultaneous probability statements). The probability statement in Theorem S5 is for a fixed finite heuristic set $\mathcal{H}$. For a countable expanding sequence $\{\mathcal{H}_k\}_{k \geq 1}$, simultaneous validity follows by assigning confidence levels $\delta_k = \delta/2^k$ and applying the union bound. This replaces $\log(2/\delta)$ by $\log(2/\delta_k) = \log(2^{k+1}/\delta)$ in the statistical term for the k -th set, while leaving the monotone supervision-bias terms unchanged.

C. Reducing Supervision Bias That Persists Regardless of Training-Set Size

Section II B contracts the supervision-bias terms by expanding the heuristic set. This subsection isolates a structurally distinct mechanism, whereby topological-consistency regularization imposes a bounded-variation constraint motivated by the degradation law of collapse distance under single-node removal (Theorem S1). Crucially, the bias component targeted by this regularization persists in the infinite-sample limit under any fixed set of heuristic labels, and therefore cannot be removed by increasing the training-set size alone. To expose the mechanism cleanly, we analyze a population-level convex hard-constraint surrogate of the implemented soft TCR penalty.

In the actual TCR-GIN objective, the regularizer is implemented as a soft penalty. The theorem below does not claim that finite-sample non-convex training with the soft penalty is exactly an orthogonal projection. Instead, it analyzes the population-level convex relaxation in which the predictor is restricted to the bounded-variation class implied by Theorem S1. This surrogate cleanly exposes why enforcing bounded variation can reduce the upward bias induced by upper-bound supervision.

We reuse the notation from Sections II A and II B. Let $N(G) = |V(G)|$. Define the convex bounded-variation class

$$\mathcal{F}_L := \left\{ f \in L^2(\mathcal{D}) : 0 \leq f(G) \leq 1, 0 \leq f(G) - \frac{N(G) - 1}{N(G)} f(G \setminus \{v\}) \leq \frac{1}{N(G)}, \forall v \in V(G), \mathcal{D}\text{-a.s.} \right\}. \quad (73)$$

This relaxes the discrete jump set $\{0, 1/N(G)\}$ in Theorem S1 to its convex hull $[0, 1/N(G)]$ and includes the pointwise box constraint $f(G) \in [0, 1]$, making $\mathcal{F}_L$ the intersection of a countable

family of closed half-spaces in $L^2(\mathcal{D})$ and hence closed and convex. Theorem S1 and the fact that $D_\tau(G) \in [0, 1]$ together imply $D_\tau \in \mathcal{F}_L$.

Let

$$f_0^*(G) := \mathbb{E}[\tilde{D}_\tau(G; \mathcal{H}) \mid G] = D_\tau(G) + \mathbb{E}[\eta(G; \mathcal{H}) \mid G] \quad (74)$$

be the unconstrained Bayes regressor of the observed upper-bound labels. Since $\tilde{D}_\tau(G; \mathcal{H}) \in [0, 1]$ for every G , the conditional expectation satisfies $f_0^*(G) \in [0, 1]$ as well. Let f_L^* be the constrained population minimizer

$$f_L^* \in \mathcal{F}_L, \quad \mathbb{E}[(f_L^* - \tilde{D}_\tau)^2] = \inf_{f \in \mathcal{F}_L} \mathbb{E}[(f - \tilde{D}_\tau)^2], \quad (75)$$

whose existence and $\mathcal{D}$ -a.s. uniqueness follow from the Hilbert projection theorem applied to the closed convex subset $\mathcal{F}_L \subset L^2(\mathcal{D})$ and the strictly convex squared-loss functional.

Theorem S6 (Improvement by bounded-variation regularization). *Under the convex hard-constraint surrogate above, f_L^* coincides with the $L^2(\mathcal{D})$ -orthogonal projection of f_0^* onto $\mathcal{F}_L$, written $f_L^* = \Pi_{\mathcal{F}_L}(f_0^*)$. Moreover, the true-label MAE satisfies*

$$\mathcal{E}_{\text{true}}(f_L^*) \leq \sqrt{\mathcal{E}_{\text{true}}(f_0^*) - \mathbb{E}[(f_0^*(G) - f_L^*(G))^2]}. \quad (76)$$

The improvement gap $\mathbb{E}[(f_0^(G) - f_L^*(G))^2]$ is non-negative, and is strictly positive whenever $f_0^* \notin \mathcal{F}_L$, in which case the right-hand side of Equation 76 is strictly smaller than $\sqrt{\mathcal{E}_{\text{true}}(f_0^*)}$.*

Proof. The proof proceeds in four steps: objective reduction to a squared distance from f_0^* , the Hilbert variational characterization of the resulting projection, a Pythagorean expansion of the true-label MSE, and conversion to a true-label MAE bound.

Step 1. Objective reduction to a squared distance. For every $f \in L^2(\mathcal{D})$, insert and subtract f_0^* inside the squared loss and expand:

$$\begin{aligned} \mathbb{E}[(f - \tilde{D}_\tau)^2] &= \mathbb{E}[(f - f_0^*)^2] + \mathbb{E}[(f_0^* - \tilde{D}_\tau)^2] \\ &\quad + 2 \mathbb{E}[(f - f_0^*)(f_0^* - \tilde{D}_\tau)]. \end{aligned} \quad (77)$$

By the defining property of the conditional expectation, $f_0^*(G) = \mathbb{E}[\tilde{D}_\tau(G; \mathcal{H}) \mid G]$, so the residual $\tilde{D}_\tau - f_0^*$ has conditional mean zero given G . Since $f - f_0^*$ is $\sigma(G)$ -measurable, the tower property gives

$$\mathbb{E}[(f - f_0^*)(\tilde{D}_\tau - f_0^*)] = \mathbb{E}[(f - f_0^*) \mathbb{E}[\tilde{D}_\tau - f_0^* \mid G]] = 0. \quad (78)$$

Therefore the cross-term in Equation 77 vanishes and

$$\mathbb{E}[(f - \tilde{D}_\tau)^2] = \mathbb{E}[(f - f_0^\star)^2] + \mathbb{E}[(f_0^\star - \tilde{D}_\tau)^2]. \quad (79)$$

The second term is independent of f . Restricting Equation 79 to $f \in \mathcal{F}_L$ and minimizing therefore reduces Equation 75 to

$$\inf_{f \in \mathcal{F}_L} \|f - f_0^\star\|_{L^2(\mathcal{D})}^2, \quad (80)$$

which is, by definition, the squared distance from $f_0^\star$ to the closed convex set $\mathcal{F}_L$. Since $\mathcal{F}_L$ is a nonempty closed convex subset of the Hilbert space $L^2(\mathcal{D})$, the Hilbert projection theorem guarantees that the minimizer exists, is unique, and equals the orthogonal projection: $f_L^\star = \Pi_{\mathcal{F}_L}(f_0^\star)$.

Step 2. Variational characterization. The $L^2(\mathcal{D})$ -projection onto a closed convex subset of a Hilbert space is uniquely characterized by the obtuse-angle condition: for every $g \in \mathcal{F}_L$,

$$\langle f_0^\star - f_L^\star, g - f_L^\star \rangle_{L^2(\mathcal{D})} \leq 0. \quad (81)$$

Since $D_\tau \in \mathcal{F}_L$ by Theorem S1 and the box constraint, specializing Equation 81 to $g = D_\tau$ yields

$$\langle f_0^\star - f_L^\star, D_\tau - f_L^\star \rangle_{L^2(\mathcal{D})} \leq 0. \quad (82)$$

Step 3. Pythagorean expansion of the true-label MSE. Expand $\|f_0^\star - D_\tau\|^2$ by inserting the intermediate point $f_L^\star$ and using the bilinearity of the inner product:

$$\begin{aligned} \|f_0^\star - D_\tau\|^2 &= \|(f_0^\star - f_L^\star) + (f_L^\star - D_\tau)\|^2 \\ &= \|f_0^\star - f_L^\star\|^2 + \|f_L^\star - D_\tau\|^2 + 2\langle f_0^\star - f_L^\star, f_L^\star - D_\tau \rangle. \end{aligned} \quad (83)$$

By Equation 82, $\langle f_0^\star - f_L^\star, D_\tau - f_L^\star \rangle \leq 0$, which is equivalent to $\langle f_0^\star - f_L^\star, f_L^\star - D_\tau \rangle \geq 0$. Hence the last term in Equation 83 is non-negative, and

$$\|f_L^\star - D_\tau\|^2 \leq \|f_0^\star - D_\tau\|^2 - \|f_0^\star - f_L^\star\|^2. \quad (84)$$

Recognizing $\mathcal{K}_{\text{true}}(f) = \|f - D_\tau\|_{L^2(\mathcal{D})}^2$, Equation 84 reads

$$\mathcal{K}_{\text{true}}(f_L^\star) \leq \mathcal{K}_{\text{true}}(f_0^\star) - \mathbb{E}[(f_0^\star(G) - f_L^\star(G))^2]. \quad (85)$$

Step 4. Conversion to a true-label MAE bound. We chain two standard inequalities for $[0, 1]$ -valued residuals. First, since $f_L^\star(G) \in [0, 1]$ (by the box constraint in $\mathcal{F}_L$) and $D_\tau(G) \in [0, 1]$, the

residual $|f_L^*(G) - D_\tau(G)|$ takes values in $[0, 1]$. Jensen's inequality applied to the concave function $\sqrt{\cdot}$ gives

$$\mathcal{E}_{\text{true}}(f_L^*) = \mathbb{E}|f_L^*(G) - D_\tau(G)| \leq \sqrt{\mathbb{E}[(f_L^*(G) - D_\tau(G))^2]} = \sqrt{\mathcal{K}_{\text{true}}(f_L^*)}. \quad (86)$$

Second, since $f_0^*(G) \in [0, 1]$ and $D_\tau(G) \in [0, 1]$, the residual $|f_0^*(G) - D_\tau(G)|$ also takes values in $[0, 1]$. Applying the pointwise inequality $t^2 \leq t$ for $t \in [0, 1]$ and integrating gives

$$\mathcal{K}_{\text{true}}(f_0^*) = \mathbb{E}[(f_0^*(G) - D_\tau(G))^2] \leq \mathbb{E}|f_0^*(G) - D_\tau(G)| = \mathcal{E}_{\text{true}}(f_0^*). \quad (87)$$

Substituting Equation 87 into the right-hand side of Equation 85 and then applying Equation 86 to the left-hand side yields

$$\mathcal{E}_{\text{true}}(f_L^*) \leq \sqrt{\mathcal{K}_{\text{true}}(f_L^*)} \leq \sqrt{\mathcal{K}_{\text{true}}(f_0^*) - \mathbb{E}[(f_0^* - f_L^*)^2]} \leq \sqrt{\mathcal{E}_{\text{true}}(f_0^*) - \mathbb{E}[(f_0^* - f_L^*)^2]}, \quad (88)$$

where the argument of the outermost square root is non-negative because $\mathcal{K}_{\text{true}}(f_L^*) \geq 0$ guarantees, via Equation 85, that $\mathcal{K}_{\text{true}}(f_0^*) \geq \mathbb{E}[(f_0^* - f_L^*)^2]$, and Equation 87 then gives $\mathcal{E}_{\text{true}}(f_0^*) \geq \mathcal{K}_{\text{true}}(f_0^*) \geq \mathbb{E}[(f_0^* - f_L^*)^2]$. This establishes Equation 76.

Strictness. The improvement gap $\mathbb{E}[(f_0^* - f_L^*)^2] = \|f_0^* - f_L^*\|_{L^2(\mathcal{D})}^2$ is zero if and only if $f_0^* = f_L^*$ $\mathcal{D}$ -a.s., which holds if and only if $f_0^* \in \mathcal{F}_L$ by uniqueness of the projection. When $f_0^* \notin \mathcal{F}_L$, the gap is strictly positive, and the right-hand side of Equation 76 satisfies $\sqrt{\mathcal{E}_{\text{true}}(f_0^*) - \mathbb{E}[(f_0^* - f_L^*)^2]} < \sqrt{\mathcal{E}_{\text{true}}(f_0^*)}$ by strict monotonicity of $\sqrt{\cdot}$. $\square$

Corollary S2 (True-label MSE improvement). *Under the hypotheses of Theorem S6, the true-label MSE satisfies*

$$\mathcal{K}_{\text{true}}(f_L^*) \leq \mathcal{K}_{\text{true}}(f_0^*) - \mathbb{E}[(f_0^*(G) - f_L^*(G))^2]. \quad (89)$$

The improvement gap is non-negative and is strictly positive whenever $f_0^ \notin \mathcal{F}_L$.*

Proof. Equation 89 is Equation 85, established in Step 3 of the proof of Theorem S6. The strictness statement follows from the same projection-uniqueness argument given at the end of that proof. $\square$

Remark S4 (Comparing the MAE and MSE improvements). The MSE improvement in Corollary S2 is a direct Pythagorean identity in $L^2(\mathcal{D})$ and provides a strict comparison $\mathcal{K}_{\text{true}}(f_L^*) < \mathcal{K}_{\text{true}}(f_0^*)$ whenever $f_0^* \notin \mathcal{F}_L$. The MAE bound in Theorem S6 passes through two intermediate steps (Jensen's inequality and the $[0, 1]$ -range bound $\mathcal{K}_{\text{true}} \leq \mathcal{E}_{\text{true}}$), so it takes the form $\mathcal{E}_{\text{true}}(f_L^*) \leq \sqrt{\mathcal{E}_{\text{true}}(f_0^*) - \Delta}$ with $\Delta := \mathbb{E}[(f_0^* - f_L^*)^2] > 0$. This is strictly tighter than $\sqrt{\mathcal{E}_{\text{true}}(f_0^*)}$ but does not by itself imply $\mathcal{E}_{\text{true}}(f_L^*) < \mathcal{E}_{\text{true}}(f_0^*)$, because the conversion from MSE to MAE is lossy. In practice, the empirical MAE improvement is observed alongside the MSE improvement, and the MSE guarantee of Corollary S2 provides the unconditional strict comparison.

Remark S5 (When is the projection improvement strictly positive?). The improvement gap $\|f_0^* - f_L^*\|^2$ is strictly positive precisely when $f_0^* \notin \mathcal{F}_L$. By Equation 74, the unconstrained Bayes regressor decomposes as $f_0^* = D_\tau + \mathbb{E}[\eta(\cdot; \mathcal{H}) \mid G]$. The first summand lies in $\mathcal{F}_L$ by Theorem S1, so the membership question reduces to whether the conditional bias $G \mapsto \mathbb{E}[\eta(G; \mathcal{H}) \mid G]$ inherits the bounded-variation structure of $\mathcal{F}_L$. Since $\tilde{D}_\tau(G; \mathcal{H}) = \min_{h \in \mathcal{H}} \hat{D}^h(G)$ is a pointwise minimum over $|\mathcal{H}|$ heuristic estimates and the heuristic that realizes this minimum generically switches under single-node perturbations, the conditional bias generically violates the single-node bounded-variation constraint on a set of positive $\mathcal{D}$ -measure, making the projection improvement strict in practice.

Remark S6 (Relation to the implemented soft TCR penalty). The implemented TCR penalty is a soft, non-convex regularizer that encourages the scale-corrected prediction change under single-node removal to follow the bounded-variation law of Theorem S1. Theorem S6 analyzes a convex hard-constraint surrogate rather than the exact finite-sample training objective. It plays a mechanistic role, showing that enforcing the bounded-variation structure satisfied by D_τ can reduce the upward bias of the observed-label Bayes regressor, as measured by both true-label MAE (Theorem S6) and true-label MSE (Corollary S2). The quantitative effect of the implemented soft penalty is therefore assessed empirically.

Remark S7 (Why this is not replaced by more samples under fixed labels). For a fixed biased observed-label mechanism $\tilde{D}_\tau(G; \mathcal{H})$, increasing the sample size drives squared-loss training toward the observed-label Bayes regressor f_0^* , not toward D_τ itself. If $f_0^* = D_\tau + \mathbb{E}[\eta \mid G]$ lies outside $\mathcal{F}_L$, bounded-variation regularization supplies structural information that is absent from the labels alone. This is the sense in which TCR addresses a residual bias that sample-size increase under a fixed biased supervision mechanism cannot remove.

Theorem S6 and Corollary S2 thus establish an improvement mechanism that is structurally distinct from the heuristic-pool expansion of Section II B. The regularization channel (Theorem S6 and Corollary S2) targets the model-side bias. Even when the label-side bias $\mathbb{E}[\eta(\cdot; \mathcal{H})]$ is fixed, bounded-variation regularization projects the ERM limit f_0^* toward a function class $\mathcal{F}_L \ni D_\tau$, strictly reducing $\mathcal{K}_{\text{true}}$ by the projection gap $\|f_0^* - f_L^*\|^2$. The supervision channel (Theorems S4 and S5) contracts the label-side supervision-bias terms $\mathbb{E}[\eta^2]$ and $\mathbb{E}[\eta]$ as the dismantling algorithm pool evolves. Neither channel can substitute for the other, because the regularization channel supplies a structural law that no amount of data under fixed labels can provide, whereas the supervision channel requires a stronger heuristic pool to reduce the label-side bias.

D. Approximate Additivity of TCR-GIN Predictions

Collapse Distance exhibits a physically grounded form of scale-transformed linear additivity. TCR-GIN is a nonlinear estimator, so exact additivity is not automatic. We therefore prove a conditional bound on the additivity error of the learned prediction.

Let the TCR-GIN prediction be written as $\hat{D}(G) = f_\theta(\mathbf{h}_G)$, where $\mathbf{h}_G = \mathcal{P}(\mathcal{E}_\psi(G))$ is the graph-level embedding obtained from the graph encoder $\mathcal{E}_\psi$ and the pooling operator $\mathcal{P}$. In this subsection, the collapse target used to generate the labels is fixed, so the target dependence of $\hat{D}$ is suppressed for readability.

For a fragmented graph $G = \bigsqcup_{j=1}^k C_j$, let $N_j = |V(C_j)|$, $N = \sum_{j=1}^k N_j$, and $w_j = N_j/N$. We denote by $\mathbf{h}_{C_j}^{\text{ctx}}$ the pooled embedding of component C_j when the encoder is run on the full graph G :

$$\mathbf{h}_{C_j}^{\text{ctx}} := \frac{1}{N_j} \sum_{v \in C_j} \mathbf{h}_v^{(G)}. \quad (90)$$

We denote by $\mathbf{h}_{C_j}^{\text{iso}}$ the pooled embedding obtained when C_j is fed independently to the same encoder:

$$\mathbf{h}_{C_j}^{\text{iso}} := \frac{1}{N_j} \sum_{v \in C_j} \mathbf{h}_v^{(C_j)}. \quad (91)$$

Thus the isolated component prediction is $\hat{D}(C_j) = f_\theta(\mathbf{h}_{C_j}^{\text{iso}})$.

Theorem S7 (Bounded additivity error of TCR-GIN). *Let*

$$\mathbf{r} := \mathbf{h}_G - \sum_{j=1}^k w_j \mathbf{h}_{C_j}^{\text{ctx}}, \quad \epsilon_j := \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_{C_j}^{\text{iso}}\|_2. \quad (92)$$

Let $\mathcal{C}$ be the convex hull of $\{\mathbf{h}_G, \mathbf{h}_{C_1}^{\text{ctx}}, \dots, \mathbf{h}_{C_k}^{\text{ctx}}, \mathbf{h}_{C_1}^{\text{iso}}, \dots, \mathbf{h}_{C_k}^{\text{iso}}\}$, and assume that f_θ is twice continuously differentiable on an open set containing $\mathcal{C}$. Define the local gradient and curvature constants

$$L_{\mathcal{C}} := \sup_{\mathbf{x} \in \mathcal{C}} \|\nabla f_\theta(\mathbf{x})\|_2, \quad \beta_{\mathcal{C}} := \sup_{\mathbf{x} \in \mathcal{C}} \|\nabla^2 f_\theta(\mathbf{x})\|_2. \quad (93)$$

Then the TCR-GIN additivity error admits the finite upper bound

$$\left| \hat{D}(G) - \sum_{j=1}^k w_j \hat{D}(C_j) \right| \leq L_{\mathcal{C}} \|\mathbf{r}\|_2 + \frac{\beta_{\mathcal{C}}}{2} \sum_{j=1}^k w_j \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2 + L_{\mathcal{C}} \sum_{j=1}^k w_j \epsilon_j. \quad (94)$$

Proof. In the TCR-GIN architecture, the prediction head f_θ is an MLP employing GeLU activation functions. Since $\text{GeLU}(x) = x \Phi(x)$ (where Φ denotes the standard Gaussian CDF) is infinitely differentiable, the composition f_θ is C^∞ , which satisfies the twice-continuous-differentiability condition of the theorem. Moreover, the set $\mathcal{C}$ is the convex hull of finitely many points in $\mathbb{R}^d$

and is therefore compact. By the extreme value theorem, the continuous functions $\|\nabla f_\theta(\cdot)\|_2$ and $\|\nabla^2 f_\theta(\cdot)\|_2$ attain their maxima on $\mathcal{C}$, so $L_{\mathcal{C}} < \infty$ and $\beta_{\mathcal{C}} < \infty$ hold automatically.

Step 1. Decomposition and bounding the context-isolation term.

We insert the contextual component predictions as an intermediate bridge. Recalling that $\hat{D}(G) = f_\theta(\mathbf{h}_G)$ and $\hat{D}(C_j) = f_\theta(\mathbf{h}_{C_j}^{\text{iso}})$, we add and subtract $\sum_j w_j f_\theta(\mathbf{h}_{C_j}^{\text{ctx}})$ inside the absolute value and apply the triangle inequality to obtain

$$\left| f_\theta(\mathbf{h}_G) - \sum_{j=1}^k w_j f_\theta(\mathbf{h}_{C_j}^{\text{iso}}) \right| \leq \underbrace{\left| f_\theta(\mathbf{h}_G) - \sum_{j=1}^k w_j f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) \right|}_{\text{Term A}} + \underbrace{\sum_{j=1}^k w_j \left| f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) - f_\theta(\mathbf{h}_{C_j}^{\text{iso}}) \right|}_{\text{Term B}}. \quad (95)$$

Term A measures how well the full-graph prediction decomposes into contextual component predictions. Term B measures how much the component predictions change when switching from contextual to isolated evaluation. We now bound Term B. For each component j , both $\mathbf{h}_{C_j}^{\text{ctx}}$ and $\mathbf{h}_{C_j}^{\text{iso}}$ belong to $\mathcal{C}$. Since $\mathcal{C}$ is convex, the entire line segment connecting these two points lies within $\mathcal{C}$. Because f_θ is differentiable on $\mathcal{C}$, the mean-value theorem guarantees the existence of a point $\boldsymbol{\xi}_j$ on this segment such that

$$f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) - f_\theta(\mathbf{h}_{C_j}^{\text{iso}}) = \nabla f_\theta(\boldsymbol{\xi}_j)^\top (\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_{C_j}^{\text{iso}}). \quad (96)$$

Taking absolute values and applying the Cauchy–Schwarz inequality gives

$$\left| f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) - f_\theta(\mathbf{h}_{C_j}^{\text{iso}}) \right| \leq \|\nabla f_\theta(\boldsymbol{\xi}_j)\|_2 \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_{C_j}^{\text{iso}}\|_2 \leq L_{\mathcal{C}} \epsilon_j, \quad (97)$$

where the last inequality uses $\boldsymbol{\xi}_j \in \mathcal{C}$ and the definition of $L_{\mathcal{C}}$. Summing over all components with weights w_j yields

$$\text{Term B} \leq L_{\mathcal{C}} \sum_{j=1}^k w_j \epsilon_j. \quad (98)$$

Step 2. Barycentric analysis and Taylor expansion for Term A.

We first examine the relationship between the full-graph embedding and the contextual component embeddings. The node set of G satisfies $V(G) = \bigcup_{j=1}^k V(C_j)$ with $|V(G)| = N = \sum_j N_j$. When the pooling operator $\mathcal{P}$ is global average pooling, the full-graph embedding decomposes as

$$\mathbf{h}_G = \frac{1}{N} \sum_{v \in V(G)} \mathbf{h}_v^{(G)} = \frac{1}{N} \sum_{j=1}^k \sum_{v \in C_j} \mathbf{h}_v^{(G)} = \sum_{j=1}^k \frac{N_j}{N} \underbrace{\left(\frac{1}{N_j} \sum_{v \in C_j} \mathbf{h}_v^{(G)} \right)}_{\mathbf{h}_{C_j}^{\text{ctx}}} = \sum_{j=1}^k w_j \mathbf{h}_{C_j}^{\text{ctx}}. \quad (99)$$

This is an algebraic identity showing that $\mathbf{h}_G$ is exactly the weighted barycenter of the contextual component embeddings. In practice, mechanisms such as virtual-node communication, batch normalization, or other global feature aggregation layers may introduce deviations from this ideal decomposition, which are captured by the barycentric residual $\mathbf{r} = \mathbf{h}_G - \sum_j w_j \mathbf{h}_{C_j}^{\text{ctx}}$.

We now expand each $f_\theta(\mathbf{h}_{C_j}^{\text{ctx}})$ around $\mathbf{h}_G$ using a second-order Taylor expansion with Lagrange remainder. Since f_θ is C^∞ , Taylor's theorem applies and gives

$$f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) = f_\theta(\mathbf{h}_G) + \nabla f_\theta(\mathbf{h}_G)^\top (\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G) + R_j, \quad (100)$$

where the second-order remainder takes the Lagrange form

$$R_j = \frac{1}{2} (\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G)^\top \nabla^2 f_\theta(\mathbf{z}_j) (\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G), \quad (101)$$

and $\mathbf{z}_j = \mathbf{h}_G + \alpha_j (\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G)$ for some $\alpha_j \in (0, 1)$ is a point on the line segment connecting $\mathbf{h}_G$ and $\mathbf{h}_{C_j}^{\text{ctx}}$. Since both endpoints belong to the convex set $\mathcal{C}$, we have $\mathbf{z}_j \in \mathcal{C}$. Applying the spectral-norm inequality $|\mathbf{v}^\top \mathbf{A} \mathbf{v}| \leq \|\mathbf{A}\|_2 \|\mathbf{v}\|_2^2$ yields

$$|R_j| \leq \frac{1}{2} \|\nabla^2 f_\theta(\mathbf{z}_j)\|_2 \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2 \leq \frac{\beta_{\mathcal{C}}}{2} \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2. \quad (102)$$

Multiplying both sides of Equation equation 100 by w_j and summing over $j = 1, \dots, k$ gives

$$\begin{aligned} \sum_{j=1}^k w_j f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) &= \underbrace{\left(\sum_{j=1}^k w_j \right)}_{=1} f_\theta(\mathbf{h}_G) + \nabla f_\theta(\mathbf{h}_G)^\top \underbrace{\left(\sum_{j=1}^k w_j \mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G \right)}_{=-\mathbf{r}} + \sum_{j=1}^k w_j R_j \\ &= f_\theta(\mathbf{h}_G) - \nabla f_\theta(\mathbf{h}_G)^\top \mathbf{r} + \sum_{j=1}^k w_j R_j. \end{aligned} \quad (103)$$

Rearranging, taking absolute values, and applying the triangle inequality together with the Cauchy–Schwarz inequality gives

$$\begin{aligned} \text{Term A} &= \left| f_\theta(\mathbf{h}_G) - \sum_{j=1}^k w_j f_\theta(\mathbf{h}_{C_j}^{\text{ctx}}) \right| = \left| \nabla f_\theta(\mathbf{h}_G)^\top \mathbf{r} - \sum_{j=1}^k w_j R_j \right| \\ &\leq |\nabla f_\theta(\mathbf{h}_G)^\top \mathbf{r}| + \sum_{j=1}^k w_j |R_j| \leq \|\nabla f_\theta(\mathbf{h}_G)\|_2 \|\mathbf{r}\|_2 + \frac{\beta_{\mathcal{C}}}{2} \sum_{j=1}^k w_j \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2 \\ &\leq L_{\mathcal{C}} \|\mathbf{r}\|_2 + \frac{\beta_{\mathcal{C}}}{2} \sum_{j=1}^k w_j \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2, \end{aligned} \quad (104)$$

where the final step uses $\mathbf{h}_G \in \mathcal{C}$ and the definition of $L_{\mathcal{C}}$.

Step 3. Combining all contributions.

Substituting the bounds on Term A (Equation equation 104) and Term B (Equation equation 98) into the decomposition of Equation equation 95 yields

$$\left| \hat{D}(G) - \sum_{j=1}^k w_j \hat{D}(C_j) \right| \leq L_C \|\mathbf{r}\|_2 + \frac{\beta_C}{2} \sum_{j=1}^k w_j \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2 + L_C \sum_{j=1}^k w_j \epsilon_j, \quad (105)$$

which is the claimed bound equation 94. This completes the proof. $\square$

Remark S8 (Interpretation). The three terms in Equation equation 94 admit clear architectural and physical interpretations.

1. **Barycentric residual** $L_C \|\mathbf{r}\|_2$ measures the failure of the full-graph embedding to equal the weighted barycenter of contextual component embeddings. This term vanishes when the readout is pure global average pooling without additional global operations.
2. **Curvature penalty** $\frac{\beta_C}{2} \sum_j w_j \|\mathbf{h}_{C_j}^{\text{ctx}} - \mathbf{h}_G\|_2^2$ is the local curvature penalty of the prediction head on the embedding convex hull. The topology-consistency regularization $\mathcal{L}_{\text{consist}}$ penalizes sharp variations on the prediction landscape, thereby implicitly suppressing β_C . Simultaneously, the GNN encoder naturally maps topologically similar components to proximal regions of the embedding space, reducing the embedding variance.
3. **Context-isolation discrepancy** $L_C \sum_j w_j \epsilon_j$ measures how much component embeddings change when components are evaluated in isolation rather than inside the full fragmented graph. Virtual-node communication, batch normalization, and global feature aggregation can all contribute to this term.

These mechanisms, namely curvature suppression and embedding concentration, synergize with the mean-field stabilization to enable TCR-GIN to progressively approximate linear additivity throughout training.

E. Computational Complexity Analysis

We analyze the forward-pass inference complexity of TCR-GIN under the assumption that node features have already been constructed. The cost of constructing the structural node features belongs to the preprocessing pipeline and is discussed separately in Supplementary Note V D.

Consider a graph $G = (V, E)$ with $N = |V|$ nodes and $M = |E|$ edges. Let L denote the number of message-passing layers and d the hidden feature dimension. A single TCR-GIN forward

pass consists of three main computational components. First, the GIN backbone performs sparse message passing over graph edges, which contributes $O(LMd)$ across L layers. The node-wise MLP or linear transformations within the GIN layers are absorbed into the $O(NLd^2)$ term below. Second, the virtual-node mechanism is implemented via global aggregation and broadcasting rather than by explicitly materializing all virtual edges, thereby incurring $O(LNd)$ overhead. Third, the LSTM-based multi-scale aggregation module processes the layer-wise embedding sequence for each node, incurring $O(NLd^2)$ time when the LSTM hidden dimension is on the order of d . Therefore, the total forward-pass inference complexity is $O(LMd + LNd + NLd^2)$. For fixed architecture dimensions L and d , this reduces to $O(N + M)$. On connected graphs, or more generally on graph families satisfying $M = \Omega(N)$, we have $N = O(M)$ and the same expression can be written as $O(M)$. We therefore describe TCR-GIN as having sparse linear forward-pass complexity with respect to graph size. This statement refers to the model forward pass and excludes preprocessing costs such as structural feature construction, data loading, and model loading.

The topological-consistency regularizer is not evaluated during inference. During training, if K node-deleted perturbations are sampled for each anchor graph, the model is evaluated on the anchor graph and on K perturbed graphs. Ignoring the small reduction in edge count after single-node deletion, the training-time forward cost of the consistency term scales as $O((K + 1)(LMd + LNd + NLd^2))$. For fixed L and d , this reduces to $O((K + 1)(N + M))$. This factor affects training only.

The inference-time memory usage is dominated by sparse graph storage and intermediate node embeddings. Sparse graph storage requires $O(N + M)$ memory, and retaining layer-wise node embeddings for multi-scale aggregation requires $O(NLd)$ memory. Excluding model parameters, the inference-time space complexity is therefore $O(N + M + NLd)$, and including model parameters adds a graph-size-independent $O(Ld^2)$ term. For fixed L and d , the space complexity reduces to $O(N + M)$.

III. ROBUSTNESS METRICS

A. Definition

Network robustness characterizes a system’s capacity to uphold structural integrity and sustain operational functionality amidst internal failures or targeted external attacks. To systematically compare existing robustness metrics, we analyze them along three key dimensions:

- **Target coupling.** Whether the metric can flexibly adapt to diverse functional maintenance

requirements, acknowledging that the definition of “maintaining function” varies across scenarios.

- **Strategy independence.** Whether the metric reflects an intrinsic property of the network itself, rather than being overly dependent on a specific removal or attack strategy.
- **Monotonicity and smoothness.** Whether the metric changes continuously and smoothly as nodes are gradually removed, enabling stable early warning rather than abrupt, noisy fluctuations.

B. Detailed Description of Main Metrics

1. Normalized size of the largest connected component (LCC size).

The size of the largest connected component is the most intuitive metric for measuring a network’s macroscopic connectivity, under the assumption that the network’s primary function is carried by its largest mutually connected component. Let N be the initial number of nodes and $|LCC|$ be the current number of nodes in the largest connected subgraph. The metric is defined as,

$$\Phi(G) = \frac{|LCC|}{N}. \quad (106)$$

In terms of *target coupling*, this metric merely mirrors the instantaneous connectivity state, failing to quantify the “safety margin” relative to a critical collapse threshold. In terms of *monotonicity and smoothness*, given that phase transitions frequently occur during the network degradation process, the LCC size is prone to precipitous, cliff-like declines. Consequently, it fails to provide stable early-warning signals due to this abrupt behavior.

2. Percolation threshold [1, 2] (p_c).

Originating from statistical physics, the percolation threshold p_c marks the critical transition point where a network shifts from a connected phase (possessing a giant component) to a fragmented phase. It is commonly used as a static predictor of the initial robustness of a network. For an infinite random network with degree distribution $P(k)$, under random failures, the Molloy–Reed criterion gives the critical removal fraction,

$$p_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}. \quad (107)$$

In terms of *target coupling*, this metric is inherently bound to the condition of “total collapse” ($N_{LCC} \approx 0$) and lacks the flexibility to accommodate scenarios requiring high connectivity

maintenance. In terms of *strategy independence*, as a theoretical construct derived largely under random failure assumptions, it reflects the statistical topology of the initial state via degree moments. In terms of *monotonicity and smoothness*, since p_c is a function of degree distribution moments ($\langle k \rangle$, $\langle k^2 \rangle$) which evolve continuously during node removal, it typically yields a smooth trajectory, avoiding abrupt jumps.

3. Natural connectivity [3] ($\bar{\lambda}$).

Rooted in spectral graph theory, natural connectivity quantifies robustness as the weighted sum of closed walks, effectively measuring path redundancy. The formula is,

$$\bar{\lambda} = \ln \left(\frac{1}{N} \sum_{i=1}^N e^{\lambda_i} \right), \quad (108)$$

where λ_i are the eigenvalues of the adjacency matrix. In terms of *target coupling*, while conceptually rigorous, its numerical abstraction poses challenges in mapping directly to tangible functional thresholds. In terms of *strategy independence*, as an intrinsic spectral attribute, its value is deterministically fixed for any given topology, remaining invariant to external algorithmic choices. In terms of *monotonicity and smoothness*, natural connectivity strictly decreases monotonically as edges are removed, and this change is smooth without sudden jumps, making it a reliable indicator of gradual degradation with no sharp transitions.

4. R-value [4].

The R-value is proposed to overcome the limitation of single-threshold metrics by considering the entire disintegration process, corresponding to the area under the LCC size degradation curve. It is defined as,

$$R = \frac{1}{N} \sum_{q=1}^N \Phi(G_q), \quad (109)$$

where G_q denotes the network after removing q nodes. In terms of *target coupling*, it mathematically represents the integration of the degradation trajectory; in engineering contexts, this integral can be truncated to align with specific collapse targets. In terms of *strategy independence*, computation of the R-value is contingent upon a specific node removal sequence (e.g., heuristic algorithms), and consequently it characterises the coupled “network–attack” system rather than the network’s intrinsic topology. In terms of *monotonicity and smoothness*, since it summarises the LCC size curve, the R-value inherits the phase-transition volatility of the underlying metric, failing to mitigate the non-stationarity of the degradation process.

IV. BASELINES

To the best of our knowledge, systematic research addressing configurable thresholds and real-time, end-to-end collapse distance estimation remains scarce. However, the prediction of collapse distance exhibits fundamental theoretical parallels with the “Network Dismantling” problem in complex network robustness. Network dismantling aims to find an optimal set or sequence of nodes such that removing these nodes will cause the network to break down at the fastest rate (usually referring to the fragmentation of the largest connected component). In contrast, collapse distance focuses on quantifying the proportion of nodes removed during this process. Given this similarity in objectives, we adopt classic and cutting-edge algorithms from the network dismantling field as baselines for estimating collapse distance. The baselines we selected include 27 advanced dismantling algorithms from the survey [5], as well as the DomiRank algorithm proposed by Engsig *et al.* [6].

To elucidate the distinct characteristics of these methods, we stratify the baselines into four primary categories based on their underlying design paradigms, namely centrality-based methods, mathematical programming and spectral analysis, heuristic methods, and machine learning methods. In the following sections, we provide a brief introduction to these algorithms, discussing their algorithm types, basic principles, specific variants, and time complexity.

A. Centrality-based Methods

These methods primarily define node importance by mining network topological characteristics and execute node removal based on score rankings. To enhance solution accuracy, such algorithms can adopt an iterative strategy to derive variants, wherein the metrics for all remaining nodes in the network are recalculated following each node removal.

1. Degree Centrality & Betweenness Centrality [7] (DC & BC).

Representing the most fundamental topological metrics, Degree Centrality targets nodes with the highest instantaneous degree, while Betweenness Centrality identifies nodes lying on the maximum number of shortest paths.

Variants: To capture the dynamic evolution of the network structure, we employ the iterative versions of these metrics. These variants, which recalculate centrality scores for all remaining nodes following each removal event, are designated as DCR (Degree Centrality Removal) and BCR (Betweenness Centrality Removal), respectively.

Time complexity: For sparse networks, DCR has a complexity of approximately $O(|V| \cdot (|V| +$

$|E|$)). BCR incurs prohibitive computational costs because it repeatedly computes all-pairs shortest paths, typically $O(|V|^3)$.

2. Collective Influence [2, 8] (CI).

Designed to transcend the local myopia of degree centrality, the CI algorithm assesses a node's influence within a spherical neighbourhood of radius ℓ . It seeks to identify the minimal node set whose removal precipitates the collapse of the giant connected component through optimal percolation.

Variants: We use three variants corresponding to search radii $\ell = 1, 2, 3$, denoted as CI ℓ -1, CI ℓ -2, and CI ℓ -3.

Time complexity: $O(|V| \log |V|)$, making it scalable for large-scale networks.

3. DomiRank [6].

This is a novel centrality metric based on node dominance. DomiRank reveals the structural vulnerability of the network by integrating local neighbour information, with the calculation involving the iterative solution of a system of linear equations. In this study, we use the standard DomiRank algorithm.

Time complexity: The complexity depends on the iteration convergence speed and the number of edges, approximately $O(T \cdot |E|)$, where T is the number of iterations.

4. Network Entanglement [9] (NE).

Based on the spectral entropy theory of complex networks, NE uses the eigenvalue spectrum of the Laplacian matrix to define the von Neumann entropy of the system. By calculating the change in the network's entanglement after node removal, this algorithm identifies key nodes. The algorithm introduces a diffusion time parameter β to examine structural features at different scales.

Variants: Based on the magnitude of β , the algorithm is categorised into small-scale (NES), medium-scale (NEM), and large-scale (NEL) variants. Furthermore, we implement extended versions incorporating a reinsertion strategy, denoted as NESR, NEMR, and NELR, respectively.

Time complexity: The algorithm involves eigenvalue decomposition; the full-spectrum computation has a complexity of approximately $O(|V|^3)$.

5. Vertex Entanglement [10] (VE).

Similar to NE, VE defines an entanglement metric at the node level and evaluates each node's contribution to the overall system spectral entropy through perturbation analysis.

Variants: This includes the standard VE version as well as a variant with a reinsertion mechanism, denoted as VER.

Time complexity: As it involves spectral computations, the complexity is relatively high, approximately $O(|V|^3)$.

B. Mathematical Programming & Spectral Analysis

This class of methods formulates network dismantling as a graph-theoretic optimisation problem (e.g., minimum-cut-style objectives) and derives approximate solutions using the graph’s algebraic or spectral properties.

1. Generalized Network Dismantling [11] (GND).

Anchored in spectral partitioning, GND utilises the Fiedler vector of the Laplacian matrix (the eigenvector associated with the second-smallest eigenvalue) to decompose the network into multiple connected subgraphs. Subsequently, the node removal process is reformulated as a weighted mathematical programming problem.

Variants: We include the standard GND and a variant with reinsertion refinement, denoted as GNDR.

Time complexity: Approximately $O(|V| \log^{2+\varepsilon} |V|)$.

2. Ensemble GND [12] (EGND).

To circumvent the susceptibility of a single spectral partitioning run to local optima, EGND employs an ensemble paradigm. It aggregates multiple GND partitioning outcomes derived from diverse initial conditions, determining the final removal sequence through consensus statistics.

Time complexity: $O(e) \cdot O(\text{GND})$, where e is the number of ensemble runs.

C. Heuristics Methods

Drawing inspiration from statistical physics models and graph-theoretic structures, these methods employ efficient iterative processes to address the dismantling problem.

1. Min-Sum [13] (MS).

Grounded in the spin-glass model of statistical physics, MS reformulates the Minimum Feedback Vertex Set (Decycling) problem as a message-passing process. By propagating probability information between nodes, it estimates the marginal probability of a node belonging to the feedback vertex set.

Variants: It includes the standard MS version and a variant with a reinsertion mechanism, denoted as MSR.

Time complexity: $O(|E|T) + O(|V|(\log |V| + T))$, where T represents the number of iterations.

2. CoreHD [14].

An efficient heuristic tailored for sparse networks, CoreHD combines k -core decomposition with a high-degree targeting strategy. The algorithm recursively prunes the network’s 2-core structure and severs long cycles, thereby avoiding computationally expensive global operations.

Variants: This study utilises the standard CoreHD algorithm.

Time complexity: Achieves near-linear complexity $O(|V|)$ on sparse networks.

3. Explosive Immunization [15] (EI).

Analogous to the phenomenon of explosive percolation, EI does not immediately seek to fragment the network but instead strategically delays the formation of the giant connected component. It applies the Clays algorithm idea, evaluating each node’s contribution to preventing the merging of the largest clusters by scoring its ability to block percolation (score σ).

Variants: This study uses two variants based on the scoring parameter: EI σ_1 ($\sigma = 1$) and EI σ_2 ($\sigma = 2$).

Time complexity: $O(|V| \log |V|)$.

D. Machine Learning Methods

Embracing a data-driven paradigm, these methods leverage Graph Neural Networks (GNNs) or Deep Reinforcement Learning (DRL) agents trained on large collections of synthetic networks to learn generalisable dismantling strategies transferable to unseen networks.

1. FINDER [16].

Operating within a deep reinforcement learning framework, FINDER frames network dismantling as a Markov Decision Process (MDP). It encodes network states using GraphSAGE and trains an agent via DQN (Deep Q-Network) to autonomously minimise the size of the largest connected component (LCC) on unseen networks.

Time complexity: The inference-stage complexity is $O(|E| + |V|(1 + \log |V|))$.

2. Graph Dismantling Machine [17] (GDM).

GDM synergises Graph Attention Networks (GAT) with geometric deep learning, casting dismantling as a node regression task. The neural network predicts each node’s importance (Criticality Score), enabling node removal based on the predicted ranking.

Variants: We examine three concrete versions. *GDM*: the standard version using a GNN to

score nodes. *GDMR*: GDM augmented with a reinsertion stage to refine the final dismantling set. *CoreGDM* [18]: an enhanced version that integrates CoreHD preprocessing (retaining only the 2-core) before neural processing, substantially reducing the search space and enabling scalability to larger networks.

Time complexity: Inference complexity scales linearly with network size and model parameters, $O(h(|V| + |E|))$, where h is the number of attention heads.

Note. Throughout this section, $|V|$ and $|E|$ denote the number of nodes and edges, respectively. Regarding the reinsertion mechanism: in algorithms such as MSR, GNDR, GDMR, NESR, VER, *etc.*, the suffix “R” indicates an added reinsertion optimisation phase. This phase uses a greedy strategy to add back nodes whose removal contributes relatively little to connectivity disruption, thereby reducing the size of the dismantling set while maintaining dismantling effectiveness.

V. DETAILS ON DATASET CONSTRUCTION

A. Synthetic Networks

To construct a balanced and representative training space, we generated a total of 84,000 networks using four network generative models. For Barabási–Albert (BA) scale-free networks, the number of edges attached by each newly added node, denoted as m , was sampled from $U(1, 8)$ to reproduce power-law degree characteristics. For Erdős–Rényi (ER) random networks, the edge probability was set to $p = \langle k \rangle / (N - 1)$, where the average degree $\langle k \rangle$ was sampled from $U(2, 8)$. For Watts–Strogatz (WS) small-world networks, the number of neighbors k was sampled from $U(4, 16)$, and the rewiring probability p was sampled from $U(0.01, 0.9)$. For Lancichinetti–Fortunato–Radicchi (LFR) benchmark networks, we set the degree-distribution exponent $\tau_1 \in U(2, 3)$, the community-size exponent $\tau_2 \in U(1.5, 2.5)$, and dynamically adjusted the mixing parameter μ between 0.2 and 0.65 according to network scale, so as to reflect the complexity of realistic community structures.

Regarding the sample distribution, we categorized networks into four scale regimes: small ($N < 300$), medium ($300 \leq N < 700$), large ($700 \leq N < 1100$), and extra-large ($1100 \leq N \leq 2000$). Subsequently, we further partitioned the node scale into 21 sub-intervals. The first two sub-intervals were $(0, 40)$ and $[40, 100)$, followed by intervals with a step size of 100. Within each scale sub-interval, we uniformly generated 1,000 instances for each model, thereby providing the deep learning model with a diverse and extensive set of high-dimensional topological feature samples.

B. Real-world Networks

We integrated a large-scale dataset for evaluating generalization and three infrastructure networks for in-depth case studies.

1. REDDIT-MULTI-12K [19] (REDDIT)

A social interaction network dataset consisting of 1,000 networks derived from online discussion threads, to evaluate the algorithm’s generalization ability on non-synthetic topologies. Each node represents a user, and edges represent reply interactions. These networks exhibit dense social structures ($N \in [300, 700]$, $\bar{N} = 495.3$).

2. London Transport Multiplex Aggr [20] (Transport)

An aggregated multiplex public transport network of London. Nodes represent stations, and edges represent connections via subway, overground, DLR, or bus routes. It is a benchmark for urban transport robustness ($N = 369$, $E = 430$, $\langle k \rangle = 2.33$).

3. Power-eris1176 [21] (Power)

The topology of the European high-voltage power grid (UCTE). Nodes represent generators or substations, and edges represent high-voltage transmission lines. This network is characterized by vulnerability to cascading failures ($N = 1176$, $E = 9864$, $\langle k \rangle = 16.78$).

4. Route-views [22] (Route)

A networks of internet autonomous systems (AS) from Oregon RouteViews. Nodes represent ASes, and edges represent peering relationships. It represents a large-scale, scale-free technological network ($N = 6474$, $E = 13895$, $\langle k \rangle = 4.29$).

C. Datasets for Attack Simulations

To capture the full evolution of a network from functional integrity to complete collapse, we performed fine-grained sampling on the networks used in our case studies. The sampling step size was selected according to network scale:

- BA-100, LFR-100 ($N = 100$): sampled at every step.
- Transport ($N = 369$): sampled every 5 steps.
- Power ($N = 1176$): sampled every 10 steps.

- WS-2000, ER-2000 ($N = 2000$) and Route: sampled every 20 steps. In particular, for the extremely large Route network, sampling was activated only after the size of the largest connected component, $|LCC_t|$, decayed to 2,000.

In practical applications, the topology of critical infrastructures is relatively stable over long time scales, which provides sufficient offline time for precomputation and for constructing targeted augmented datasets. However, the inference phase demands high computational efficiency to ensure effective real-time monitoring. To this end, we built a domain-adaptive augmentation set by performing random attacks on the target real networks and sampling subgraphs in a stratified manner. We applied stratified discretization to the sampling space to ensure adequate coverage across different stages of the collapse trajectory:

- Transport: partitioned into 5 uniform intervals, $N_{\text{sub}} \in \{(0, 40), [40, 100), [100, 200), [200, 300), [300, 369]\}$, with 3,000 samples per interval.
- Power: partitioned into 13 intervals. The first two intervals use a fine-grained split $((0, 40), [40, 100))$, followed by fixed-width intervals with step size 100 $([100, 200), \dots, [1100, 1176])$, with 3,000 samples per interval.
- Route: partitioned into 21 intervals. For the critical dismantling phase with $N < 1100$, we used 12 fine-grained intervals with step size 100 (including $(0, 40), [40, 100)$), with 2,000 samples per interval; for the initial degradation phase with $N \in [1100, 2000]$, we partitioned into 9 intervals, with 1,000 samples per interval.

This construction compensates for the domain gap between synthetic and real data while remaining compatible with practical operational constraints.

D. Node Structural Features

To balance computational efficiency and representational capacity, we extract node-level structural features as the initial input to the encoder. The main model primarily uses the first three basic features, while the extended features are used for sensitivity analysis (see Supplementary Note VIII). The computational costs below refer to feature construction before the TCR-GIN forward pass. These preprocessing costs are separate from the model inference complexity analyzed in Supplementary Note II E. Unless otherwise stated, we consider simple, unweighted, undirected networks. Memory costs refer to additional working/output memory beyond the $O(N + M)$ input networks storage.

1. Degree (k_v)

The most basic centrality measure, representing the number of direct connections of a node.

$$k_v = |\mathcal{N}(v)|, \quad (110)$$

where $\mathcal{N}(v)$ is the neighbor set of node v . All node degrees can be computed by one pass over the edge list, with time complexity $O(N + M)$ and $O(N)$ additional memory for the degree vector.

2. Clustering Coefficient (C_v)

Measures the extent to which a node's neighbors are connected, quantifying local cohesiveness.

$$C_v = \frac{2L_v}{k_v(k_v - 1)}, \quad (111)$$

where L_v is the number of edges that actually exist among the k_v neighbors of node v . Computing local clustering coefficients requires counting edges among the neighbors of each node. With degree-oriented neighbor-list intersections, all local triangle counts can be computed in $O\left(\sum_{(u,v) \in E} \min(d_u, d_v)\right)$ time, which is bounded by $O(M^{3/2})$ in the worst case and becomes near-linear on sparse graphs with bounded degeneracy. The additional memory required for storing the output coefficients is $O(N)$.

3. K-core Number

Defined as the largest integer k such that node v belongs to a k -core (i.e., the maximal subgraph in which every node has degree at least k). Computationally, all k -core numbers can be obtained by the standard peeling algorithm with a bucket queue, requiring $O(N + M)$ time and $O(N)$ additional memory.

4. Average Neighbor Degree ($k_{nn}(v)$)

The average degree of all neighbors of node v , used to capture assortative mixing patterns (i.e., whether high-degree nodes tend to connect to other high-degree nodes).

$$k_{nn}(v) = \frac{1}{k_v} \sum_{u \in \mathcal{N}(v)} k_u. \quad (112)$$

Computationally, given the degree vector, all average-neighbor-degree values can be computed by accumulating neighbor degrees over edges, requiring $O(N + M)$ time and $O(N)$ additional memory.

5. PageRank (PR_v)

A measure of propagating influence, computed iteratively.

$$PR_v = \frac{1 - \alpha}{N} + \alpha \sum_{u \in \mathcal{N}(v)} \frac{PR_u}{k_u^{out}}, \quad (113)$$

where $\alpha = 0.85$ is the damping factor. For T power-iteration steps, PageRank requires $O(TM)$ time and $O(N)$ additional memory.

6. Betweenness Centrality (B_v)

Quantifies the number of shortest paths that pass through the node, reflecting its role as a bridge in information flow.

$$B_v = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (114)$$

where σ_{st} is the total number of shortest paths from node s to t , and $\sigma_{st}(v)$ is the number of those paths that pass through node v . For unweighted networks, exact betweenness centrality can be computed by Brandes' algorithm in $O(NM)$ time and $O(N + M)$ additional memory.

7. Eigenvector Centrality (e_v)

This metric considers not only a node's number of connections but also the importance of its neighbors. It corresponds to the leading eigenvector of the adjacency matrix and its associated component, reflecting a node's influence gained by connecting to high-influence nodes. It is defined as:

$$e_v = \frac{1}{\lambda} \sum_{u \in \mathcal{N}(v)} e_u, \quad (115)$$

where $\mathcal{N}(v)$ denotes the neighbor set of node v , and λ is the largest eigenvalue of the adjacency matrix A . This relation indicates that a node's centrality score is proportional to the sum of the centrality scores of its neighbors. Using power iteration for T iterations, eigenvector centrality requires $O(TM)$ time and $O(N)$ additional memory.

E. Generation of True Labels and Observed Labels

To support supervised training, each network instance in the dataset is assigned a scalar label representing the collapse distance for a specific collapse target.

1. Observed Labels Based on Ensemble Baseline ($N > 40$)

Since finding the theoretically optimal dismantling sequence for a general networks is an NP-hard problem, for networks with node sizes $N > 40$, we generate high-quality observed labels based on several heuristic algorithms. Specifically, we perform 28 baseline algorithms detailed in Supplementary Note IV on each network instance. The observed label y is defined as the minimum collapse distance obtained by this set of heuristic algorithms:

$$\tilde{D}_\tau(G; \mathcal{H}) = \min_{h \in \mathcal{H}} \{y_h(G, \tau)\}, \quad (116)$$

where $\mathcal{H}$ represents the set of baseline strategies, and $y_h(G, \tau)$ represents the dismantling cost of network G under the collapse target τ as determined by algorithm h . This method provides an upper

bound on the optimal solution, ensuring that the model learns from the current best strategies.

2. True Labels Based on Exhaustive Search ($N \in [20, 40]$)

To validate the model's ability to recognize the theoretically optimal solution, we developed an exhaustive search algorithm to explore the solution space and calculate the ground truth labels for small-scale networks [16, 17] ($N \in [20, 40]$). These ground truth labels serve as the absolute standard for evaluating the model's accuracy on small networks.

VI. PERFORMANCE METRICS

We evaluate the model using a set of complementary statistical metrics to assess: (i) the estimation accuracy on each remaining network, (ii) the monotonicity and smoothness of the online output sequence required by Theorem S1, and (iii) the additivity required by Theorem S2 when the network splits into multiple connected components.

Let the sequence of remaining networks under sustained attacks be denoted by $\{G_{t_i}\}_{i=1}^L$, where $t_1 < \dots < t_L$ are the evaluated snapshot steps. Let $N_{t_i} = |V(G_{t_i})|$ be the number of nodes in the remaining network at step t_i , and denote the initial size as $N_0 = |V(G_0)|$. For each snapshot, the label is the local collapse distance $d_{t_i} := D_\tau(G_{t_i})$, i.e., the normalized fraction of additional node removals required to drive the current remaining network to collapse. The model output is $\hat{d}_{t_i} := \hat{D}_\tau(G_{t_i})$. To provide a unified characterization across different network sizes and sampling intervals, we convert the local collapse distance to the global collapse distance relative to the initial network size:

$$D_{t_i} = d_{t_i} \frac{N_{t_i}}{N_0}, \quad \hat{D}_{t_i} = \hat{d}_{t_i} \frac{N_{t_i}}{N_0}. \quad (117)$$

The evaluation on a single network can be regarded as a special case with $L = 1$. In this case, $N_{t_1} = N_0$ and thus the global collapse distance reduces to the local one, i.e., $D_{t_1} = d_{t_1}$.

A. Mean Absolute Error (MAE).

MAE measures snapshot-level estimation accuracy and applies to both single-network evaluation and online evaluation. For the sample set $\mathcal{I}$ included in the evaluation (the test graphs in single-network evaluation, or the attack snapshots included in online evaluation), we define:

$$\text{MAE} = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} \left| \hat{d}_t - d_t \right|. \quad (118)$$

A smaller MAE indicates a more accurate estimation of the collapse distance for an individual remaining network. Specifically, two MAE metrics are reported and clearly distinguished. The observed-label MAE $\text{MAE}^{\text{OL}}(f) := \mathbb{E}|f(G) - \tilde{D}_\tau(G; \mathcal{H})|$ measures agreement with the best available upper-bound supervision and is reported on the synthetic and real-world networks of all scales, because exact collapse distance is computationally infeasible there. The true-label MAE $\text{MAE}^{\text{TL}}(f) := \mathbb{E}|f(G) - D_\tau(G)|$ measures agreement with the exact collapse distance and is reported only on the small graphs of Fig. 3f and g, where exhaustive search is feasible.

B. Monotonicity violation frequency (M).

Theorem S1 theoretically indicates that the collapse distance should monotonically decrease as the attack progresses. Hence, any increase in the predicted global collapse distance sequence $\{\hat{D}_{t_i}\}$ is counted as a monotonicity violation. Define the consecutive difference

$$\Delta_i = \hat{D}_{t_{i+1}} - \hat{D}_{t_i}, \quad i = 1, \dots, L-1, \quad (119)$$

and introduce a fixed tolerance $\epsilon \geq 0$ to absorb minor fluctuations caused by numerical errors and approximate inference. The monotonicity violation frequency is then defined as

$$M = \frac{1}{L-1} \sum_{i=1}^{L-1} \mathbb{I}(\Delta_i > \epsilon), \quad (120)$$

where $\mathbb{I}(\cdot)$ is the indicator function. $M \in [0, 1]$, and a smaller M implies fewer unreasonable rebounds along the trajectory, enhancing the reliability of the output as an online early-warning signal.

C. Smoothness violation frequency (S).

Theorem S1 further suggests that the collapse distance should decrease smoothly as the attack proceeds, i.e., it should not exhibit abrupt drops exceeding the physical limit. Let the step span be $\delta t_i = t_{i+1} - t_i$. Under the setting that at most one node is removed per step, the theoretical maximum decrease in the global collapse distance (relative to the initial size) over δt_i steps is $\delta t_i / N_0$. Therefore, we regard any drop that exceeds this theoretical upper bound (with an added tolerance) as a smoothness violation:

$$S = \frac{1}{L-1} \sum_{i=1}^{L-1} \mathbb{I}\left(\Delta_i < -\left(\frac{\delta t_i}{N_0} + \epsilon\right)\right), \quad (121)$$

where $S \in [0, 1]$, and a smaller S indicates fewer step-like abrupt drops along the trajectory, which better matches the demand for stable readings in online monitoring.

D. Additivity mean absolute error (A-MAE).

During an attack, the remaining network may split into multiple connected components. Theorem S2 theoretically states that the model should be consistent across two equivalent computation paths: (i) inferring on the whole remaining network, and (ii) inferring on each component and then aggregating them with size-based weights. We therefore define an additivity error to quantify the discrepancy.

Let the connected components of the remaining network G_t be $\{\mathcal{C}_{t,j}\}_{j=1}^{k_t}$, with sizes $N_{t,j} = |V(\mathcal{C}_{t,j})|$, satisfying $\sum_j N_{t,j} = N_t$. Define the component-aggregated prediction as

$$\hat{d}_t^{\text{agg}}(G_t) = \sum_{j=1}^{k_t} \frac{N_{t,j}}{N_t} \hat{d}(\mathcal{C}_{t,j}). \quad (122)$$

We then measure the difference between the two paths in an MAE form:

$$\text{A-MAE} = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} \left| \hat{d}(G_t) - \hat{d}_t^{\text{agg}}(G_t) \right|. \quad (123)$$

A smaller A-MAE indicates that the model better preserves the linear additivity under scale transformation.

VII. MODEL CONFIGURATION

To determine the optimal model architecture and training strategy, we conducted a systematic grid search over the hyperparameter space outlined below. The final model configuration was selected based on performance metrics on the validation set.

A. Graph Encoder Architecture: TCR-GIN

- Number of GIN layers: $\{3, 4, 5\}$.
- Hidden layer dimension: $\{64, 128\}$.
- Activation function: GELU.

B. Regressor Architecture: MLP

- Number of layers: Fixed at 2 layers.
- Number of hidden neurons: Consistent with encoder dimension.
- Dropout probability: $\{0.2, 0.3, 0.5\}$.

C. Training Strategy

- Optimizer: Adam.
- Learning rate: $\{10^{-3}, 3 \times 10^{-4}, 10^{-4}\}$ (with cosine annealing decay).
- L2 regularization (weight decay): $\{5 \times 10^{-4}, 10^{-4}, 10^{-5}\}$.
- Consistency loss weight (λ): $\{0, 0.2, 0.5, 0.8\}$.
- Batch size: $\{32, 64, 128\}$.
- Number of epochs: $\{200\}$.
- Label-scaling factor: $s \in \{1, 10, 100\}$. During training, the observed labels were multiplied by a scaling factor s to improve numerical conditioning, since the collapse distances can be very small and scaling them up facilitates gradient-based optimization. During inference and testing, the model outputs were divided by s to recover the original scale.

VIII. ABLATION STUDIES AND SENSITIVITY ANALYSIS

This section presents the structural ablation results of TCR-GIN, as well as the sensitivity analysis of three key hyperparameters (λ , K , and d_{in}). We evaluate in two settings. The first setting involves synthetic and real network predictions, including four different sizes of synthetic networks (Small, Medium, Large, Extra-large) and one real network (REDDIT), with the metric being MAE (mean $\pm$ std). The second setting involves predictions under attack scenarios, which include six networks: BA100, LFR100, ER2000, WS2000, Transport, and Power. In addition to MAE, we also report the monotonicity violation frequency M and the smoothness violation frequency S . The values of M and S are scaled by a factor of 100 in all tables and figures for readability. The first setting assesses prediction accuracy on individual networks, whereas the attack-scenario setting

further evaluates whether the variants produce stable degradation trajectories. Unless otherwise specified, the default settings for TCR-GIN are $\lambda = 0.2$, $K = 1$, and $d_{\text{in}} = 3$.

A. Structural Ablation: Multi-scale Aggregation, Residual Connections, and Virtual Node

We introduce three core designs in the TCR-GIN framework to cope with the complex topological evolution during network collapse: (i) the JK-LSTM mechanism for adaptive multi-scale feature aggregation, (ii) residual connections to facilitate gradient propagation in deeper architectures, and (iii) a virtual node to capture global information. To quantify the contribution of each component, we construct corresponding model variants and compare them on both the synthetic and real-network datasets (Table S4) and the attack scenarios (Table S5).

First, the JK-LSTM mechanism primarily enhances the model’s ability to adaptively adjust its effective receptive field, thereby addressing the structural mismatch in which some nodes require information from farther neighborhoods while others only need local neighbors. The optimal depth of message passing depends on the network structure and a node’s position within it, so a fixed depth is often not robust—this is especially evident on real networks. For example, on REDDIT, JK-LSTM achieves an MAE of 0.008 ± 0.0009 , whereas last-layer (JK-Last) aggregation degrades performance to 0.022 ± 0.0024 . In attack-scenario prediction, JK-Last aggregation also simultaneously amplifies monotonicity and smoothness violations. Taking Power as an example, JK-LSTM attains an MAE of 0.029 ± 0.0105 ; after replacing it with Last, the MAE increases to 0.082 ± 0.0621 , and both monotonicity and smoothness violation frequencies rise significantly (Table S5).

Residual connections are one of the key stabilizing components in attack-scenario prediction. Originally introduced to mitigate degradation and optimization difficulties in deep networks, residual learning stabilizes gradient flow and representation propagation by explicitly preserving an identity path [23]. In GNNs, this is even more critical because stacking multiple message-passing layers can easily lead to over-smoothing and information loss, and training is more prone to vanishing gradients. For example, on BA100, TCR-GIN achieves an MAE of 0.019 ± 0.0016 , whereas removing residual connections causes the MAE to surge to 0.584 ± 0.2849 . Meanwhile, the frequencies of monotonicity and smoothness violations increase significantly (Table S5). Similar catastrophic degradation is consistently observed on LFR100, Transport, and other networks.

Virtual nodes primarily affect the way global information is injected into the model. A virtual node functions as a global “aggregation-broadcast” channel in the graph that connects to all nodes.

It enables rapid propagation of global context within a limited number of layers, addressing the challenge of long-range dependency propagation. This characteristic explains why GIN augmented with a virtual node serves as a robust baseline in benchmarks such as the Open Graph Benchmark (OGB) [24]. In our task, the collapse distance is influenced both by locally fragile structures and by the global connectivity degradation. The virtual node helps reduce MAE on most synthetic networks and improves smoothness in attack scenarios (see Tables S4 and S5). However, it does not always bring improvements on REDDIT. This phenomenon suggests that injecting global information does not necessarily lead to stronger generalization. For networks where the information distribution is more locally determined, the virtual node may even introduce bias.

B. Sensitivity of TCR-GIN to the Consistency Regularization Weight λ

Consistency regularization constrains the model’s outputs to adhere to the theoretical change bound under single-node perturbations via a composite loss: $(1 - \lambda)\mathcal{L}_{\text{sup}} + \lambda\mathcal{L}_{\text{consist}}$. This bound is derived from the monotonicity–smoothness theorem. We evaluate $\lambda \in \{0, 0.2, 0.5, 0.8\}$ while fixing $K = 1$ and $d_{\text{in}} = 3$. The results are shown in Figure S8.

We observe a typical regularization trade-off. On static networks, MAE is generally insensitive to λ : most variations remain within the range of the error margins. This suggests that under in-distribution fitting, the supervised signal is already sufficiently strong.

In attack scenarios, a moderate λ is often more robust. It can reduce violation frequencies and, on some networks, improve MAE. For instance, on Power, moving from $\lambda = 0$ to $\lambda = 0.5$ yields reduced MAE, while an excessively large λ leads to degradation, because an over-weighted consistency term competes with the supervised objective and impedes the fitting process. On networks such as ER2000, increasing λ can also significantly suppress monotonicity violations M , aligning with the intended role of the consistency term.

Based on these results, we choose $\lambda = 0.2$ as the default. It maintains low violation frequencies across multiple networks without systematically harming MAE. Furthermore, it imposes no additional computational overhead during inference, as λ merely re-weights the loss terms and does not change the model’s forward-pass structure.

C. Sensitivity of TCR-GIN to the Sampling Size K

The consistency loss uses a worst-of- K strategy. During training, we sample K single-node deletion subgraphs from the same anchor graph and backpropagate the gradient from the subgraph with the largest consistency error. This strategy is designed to approximate the consistency constraint under the “worst-case” perturbations. We evaluate $K \in \{1, 3, 5\}$ while fixing $\lambda = 0.2$ and $d_{\text{in}} = 3$. The results are shown in Figure S9.

Overall, no single value of K consistently dominates across all datasets. Meanwhile, the computational cost of training increases with K , as a larger K requires more forward calculations for the candidate subgraphs. Because candidate subgraph generation is used only during training, the inference latency remains unaffected. Considering that the performance advantage of larger K is inconsistent and the training overhead grows substantially, we adopt $K = 1$ as the default setting.

D. Sensitivity of TCR-GIN to the Input Feature Dimension d_{in}

Our default input feature dimension is $d_{\text{in}} = 3$, consisting of degree, clustering coefficient, and k-core value. We further evaluate $d_{\text{in}} \in \{3, 5, 7\}$ while fixing $\lambda = 0.2$ and $K = 1$. Specifically, $d_{\text{in}} = 5$ augments the default set with Average Neighbor Degree and PageRank score; $d_{\text{in}} = 7$ further includes Betweenness Centrality and Eigenvector Centrality. Results are reported in Figure S10.

We observe that increasing d_{in} does not yield consistent performance gains. This indicates that additional hand-crafted features do not necessarily provide learnable information. Ultimately, whether these features help depends on whether they align with the true determinants of collapse distance. In contrast, the computational latency increases consistently and noticeably. As d_{in} grows, inference latency increases substantially, an effect that is particularly pronounced in larger-scale networks, as shown in Figure S10l. Given the marginal performance gains but significant computational overhead, we adopt $d_{\text{in}} = 3$ as the default input setting.

IX. SCALABLE DEPLOYMENT STRATEGY ON LARGE-SCALE NETWORKS

In real-world scenarios, robustness monitoring typically focuses on the high-risk interval where a system is close to collapse, rather than pursuing uniformly high precision across the entire lifecycle. Guided by this operational logic, we optimize the model to achieve superior accuracy and efficiency on networks of a moderate scale, thereby concentrating computational resources on the late-stage critical regime of larger-scale systems. Moreover, leveraging the scale-transformed linear additivity

established in Theorem S2, we theoretically establish the feasibility of decoupling the prediction for a complete large-scale system into component-level predictions for fragmented subgraphs. Specifically, we construct a high-precision estimation model tailored to networks with up to 2,000 nodes, so that while keeping training and inference costs controllable, it can provide reliable estimates of the collapse distance when a larger-scale network degrades to the critical stage. The model’s adaptation scale can be adjusted according to training coverage and deployment resources.

We adopt a conditional late-stage activation strategy for the online estimation of large-scale networks. As the attack sequence progresses, we track the size of the largest connected component, denoted as $|LCC_t|$, at each step. The step when $|LCC_t| \leq 2000$ for the first time serves as the activation threshold. From this point onward, the large-scale system is decoupled into multiple subcomponents that fall within the adaptation range. By predicting their respective collapse distances and performing scale-transformed aggregation, we can reconstruct the global metrics. Metrics such as MAE, monotonicity, and smoothness are calculated exclusively within this activation interval. This activation rule only depends on the currently observable $|LCC_t|$ and does not require any information about future collapse steps. Additionally, to ensure a fair comparison, all benchmark methods are evaluated and statistically compared within the same activation interval.

We use Route as a representative case to validate the effectiveness of this design. Route contains 6,474 nodes, and its initial scale is significantly larger than the high-precision adaptation range of the model tailored to 2,000-node networks. Figure S11 illustrates the online collapse distance trajectories of Route under 30 attack strategies. It can be observed that, within the late-stage critical regime, there are still pronounced disagreements among baseline methods: under certain strategies, step-like changes and oscillations occur, and the estimated values differ substantially in magnitude. In contrast, within the activation interval where $|LCC_t| \leq 2000$, TCR-GIN produces more continuous outputs, with better-controlled fluctuations, and maintains a monotonic degradation pattern consistent with the theoretical characterization in the main text. Across multiple types of strategies, it provides stable collapse distance estimates that better match the needs of late-stage monitoring. These results indicate that a model designed for networks with fewer than 2,000 nodes can, without changing its architecture or inference pipeline, deliver high-precision monitoring in the late-stage critical regime for much larger real-world networks, thereby meeting the joint requirements of stability and efficiency for online deployment.

These results provide a practically deployable pathway for scalable monitoring. The scale-transformed linear additivity of collapse distance extends the application range of the model beyond the network sizes directly covered during training. Although the model is trained on smaller-scale

networks, it can still be applied to a larger network once the online degradation process fragments the system into components that fall within the trained adaptation range. The global collapse-distance estimate can then be reconstructed from component-level predictions through scale-transformed aggregation. This strategy does not require modifying the model architecture or retraining the inference pipeline. It therefore focuses high-precision monitoring on the late-stage critical regime, where warning signals are most relevant for operational decision-making.

X. LIGHTWEIGHT PREDICTION OF DISCRETE COLLAPSE PROFILES

The main experiments focus on the fixed-target scalar setting, where TCR-GIN predicts the collapse distance $D_\tau(G)$ for a prescribed collapse target τ . To assess whether the same framework can be extended beyond a single operational threshold, we further implemented a lightweight discrete collapse-profile predictor.

For a given grid of collapse targets $\mathcal{T}$, which is set in our experiments to $\mathcal{T} = \{0.10 + 0.05j\}_{j=0}^{16}$, the collapse profile of a network G is defined as

$$\mathbf{D}(G) = [D_{\tau_1}(G), D_{\tau_2}(G), \dots, D_{\tau_K}(G)], \quad (124)$$

where $K = 17$. Profile labels were obtained by applying the same observed-label construction used in the scalar setting independently at each target value. Thus, for each network, the supervised target is the heuristic-derived observed collapse profile $\mathbf{D}^{\text{obs}}(G) = [D_{\tau_1}^{\text{obs}}(G), D_{\tau_2}^{\text{obs}}(G), \dots, D_{\tau_K}^{\text{obs}}(G)]$.

As in the main experiments, these labels should be interpreted as dismantling-derived upper-bound supervision rather than exact optimal collapse distances. The profile model follows the TCR-GIN architecture and training protocol described in the main text. The only architectural modification is that a K -dimensional output head replaces the final scalar regression head,

$$\hat{\mathbf{D}}(G) = [\hat{D}_{\tau_1}(G), \hat{D}_{\tau_2}(G), \dots, \hat{D}_{\tau_K}(G)]. \quad (125)$$

The profile head uses a sigmoid output activation so that each predicted collapse-distance coordinate lies in $[0, 1]$. The supervised loss is extended from scalar mean squared error to profile-wise mean squared error,

$$\mathcal{L}_{\text{sup}}^{\text{profile}} = \frac{1}{BK} \sum_{i=1}^B \sum_{k=1}^K (\hat{D}_{\tau_k}(G_i) - D_{\tau_k}^{\text{obs}}(G_i))^2, \quad (126)$$

where B is the mini-batch size. The single-node perturbation consistency regularization used in scalar TCR-GIN is applied coordinate-wise over the target grid and averaged across τ . We

additionally include a soft monotonicity prior along the target dimension. Since a larger collapse target is easier to reach, the collapse distance should be non-increasing with respect to τ ,

$$D_{\tau_1}(G) \geq D_{\tau_2}(G) \geq \dots \geq D_{\tau_K}(G), \quad \tau_1 < \tau_2 < \dots < \tau_K. \quad (127)$$

We penalize violations of this ordering using

$$\mathcal{L}_{\text{mono}} = \frac{1}{B(K-1)} \sum_{i=1}^B \sum_{k=1}^{K-1} \left[\max(0, \hat{D}_{\tau_{k+1}}(G_i) - \hat{D}_{\tau_k}(G_i)) \right]^2. \quad (128)$$

The total loss is

$$\mathcal{L} = (1 - \lambda_{\text{con}} - \lambda_{\text{mono}}) \mathcal{L}_{\text{sup}}^{\text{profile}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}^{\text{profile}} + \lambda_{\text{mono}} \mathcal{L}_{\text{mono}}. \quad (129)$$

In this experiment, we used $\lambda_{\text{con}} = 0.2$ and $\lambda_{\text{mono}} = 0.1$. All other architectural and optimization settings follow the scalar TCR-GIN setup.

We evaluated this lightweight profile extension on 100 REDDIT test networks. The primary metric is profile mean absolute error,

$$\text{MAE}_{\text{profile}} = \frac{1}{K} \sum_{k=1}^K |\hat{D}_{\tau_k}(G) - D_{\tau_k}^{\text{obs}}(G)|. \quad (130)$$

The reported value is averaged over the REDDIT test networks. TCR-GIN achieved a profile MAE of 0.0075 ± 0.0016 , indicating that the scalar collapse-distance estimator can be extended to predict a discrete target-dependent collapse profile with only a small modification to the output head and loss function. To visualize the prediction behavior, we plot the observed and predicted collapse profiles for five representative REDDIT test networks in Figure. S20. These networks were selected to span the empirical distribution of per-network profile errors rather than to show only best-case examples. Overall, the predicted profiles follow the observed target-dependent trend, supporting the feasibility of lightweight discrete collapse-profile prediction.

XI. SUPPLEMENTARY TABLES

Table S1. **Comparison of algorithmic estimation accuracy.** This table presents the prediction error (MAE) $\downarrow$ of different algorithms across various datasets. Synthetic networks are categorised by node size into Small, Medium, Large, and Extra-large (definitions provided in Supplementary Note V), while real-world evaluation uses the REDDIT dataset. For TCR-GIN, values denote the mean $\pm$ s.d. across networks and five training seeds. **Blue text** indicates the best performance (minimum error), and **gray shading** indicates the second-best performance.

Algorithm	Small	Medium	Large	Extra-large	REDDIT
TCR-GIN	0.011 $\pm$ 0.0001	0.008 $\pm$ 0.0005	0.005 $\pm$ 0.0007	0.005 $\pm$ 0.0009	0.008 $\pm$ 0.0008
DC	0.190	0.282	0.267	0.276	0.064
DCR	0.044	0.067	0.059	0.062	0.029
BC	0.167	0.292	0.252	0.266	0.129
BCR	0.027	0.030	0.034	0.035	0.002
DomiRank	0.120	0.185	0.195	0.206	0.051
FINDER	0.072	0.088	0.125	0.131	0.005
GDM	0.034	0.038	0.051	0.052	0.004
GDMR	0.013	0.015	0.020	0.020	0.001
CoreGDM	0.008	0.010	0.014	0.015	0.001
MS	0.018	0.072	0.007	0.007	0.139
MSR	0.021	0.017	0.004	0.003	0.023
GND	0.115	0.152	0.169	0.186	0.170
GNDR	0.058	0.057	0.062	0.061	0.021
CI ℓ -1	0.024	0.029	0.033	0.042	0.104
CI ℓ -2	0.098	0.140	0.128	0.133	0.167
CI ℓ -3	0.160	0.175	0.147	0.151	0.170
CoreHD	0.011	0.005	0.006	0.007	0.001
EGND	0.097	0.098	0.113	0.114	0.023
EI σ_1	0.077	0.111	0.138	0.152	0.104
EI σ_2	0.419	0.457	0.452	0.460	0.733
NES	0.215	0.305	0.289	0.297	0.075
NESR	0.050	0.049	0.050	0.050	0.017
NEM	0.176	0.326	0.252	0.265	0.191
NEMR	0.045	0.044	0.044	0.044	0.022
NEL	0.181	0.337	0.236	0.243	0.182
NELR	0.046	0.044	0.043	0.043	0.020
VE	0.157	0.342	0.246	0.259	0.891
VER	0.044	0.088	0.046	0.047	0.323

Table S2. **Comparison of computational efficiency.** This table reports the average end-to-end inference time (seconds per graph) for different algorithms under identical experimental conditions. Measurements encompass the entire pipeline, including data loading, feature construction, and model forward inference. All experiments were conducted in the same hardware environment (Intel Xeon Platinum 8470Q, 25-core, Python implementation). Each entry represents the mean inference time averaged over all graphs within the corresponding test set.

Algorithm	Small	Medium	Large	Extra-large	REDDIT
TCR-GIN	0.016	0.043	0.181	0.189	0.027
DC	0.002	0.008	0.024	0.060	0.002
DCR	0.005	0.024	0.065	0.154	0.006
BC	0.002	0.011	0.036	0.090	0.007
BCR	0.024	0.296	1.574	9.237	0.026
DomiRank	2.187	3.832	6.953	14.735	3.133
FINDER	0.273	0.764	1.788	2.804	0.223
GDM	5.075	5.436	6.392	8.635	5.219
GDMR	20.444	22.391	25.700	31.054	20.521
CoreGDM	36.258	50.992	77.411	118.457	39.234
MS	2.388	9.340	17.070	29.729	7.899
MSR	2.493	9.467	17.205	29.886	7.944
GND	0.110	0.167	0.263	0.506	0.162
GNDR	0.104	0.125	0.166	0.249	0.277
CI ℓ -1	0.419	0.479	0.593	0.711	0.376
CI ℓ -2	0.419	0.491	0.661	0.824	0.374
CI ℓ -3	0.432	0.573	1.352	1.534	0.384
CoreHD	0.336	0.362	0.416	0.474	0.297
EGND	5.395	38.801	129.229	337.041	10.079
EI σ_1	0.107	0.149	0.240	0.416	0.082
EI σ_2	0.104	0.150	0.272	0.494	0.086
NES	0.724	14.887	58.719	237.776	11.208
NESR	0.089	0.132	0.164	0.194	11.330
NEM	0.705	15.132	58.143	241.798	11.434
NEMR	0.096	0.140	0.178	0.207	11.563
NEL	0.728	15.961	57.458	242.512	11.245
NELR	0.091	0.135	0.161	0.257	11.377
VE	0.756	6.713	19.464	53.505	6.437
VER	4.109	4.135	4.165	4.335	10.624

Table S3. **Validation against ground truth labels on small-size networks.** This table compares the MAE relative to exact ground truth labels (TL) and observed labels (OL). TLs are obtained via exhaustive search, while OLs are derived from the minimum collapse distance calculated by an ensemble of 28 baseline heuristics. ‘True-label MAE’ quantifies the deviation from the theoretical optimum, whereas ‘Observed-label MAE’ measures the fit to the heuristic baseline. Since observed labels are derived from the baseline ensemble, a low Observed-label MAE does not necessarily imply a low True-label MAE. **Blue text** indicates the best performance, and **gray shading** indicates the second-best performance.

Algorithm	True-label MAE	Observed-label MAE
TCR-GIN	0.019 $\pm$ 0.0012	0.023 $\pm$ 0.0009
DC	0.049	0.016
DCR	0.107	0.075
BC	0.191	0.159
BCR	0.034	0.017
DomiRank	0.041	0.003
FINDER	0.102	0.002
GDM	0.111	0.009
GDMR	0.072	0.078
CoreGDM	0.358	0.039
MS	0.086	0.325
MSR	0.050	0.081
GND	0.036	0.039
GNDR	0.114	0.018
CI ℓ -1	0.072	0.024
CI ℓ -2	0.051	0.151
CI ℓ -3	0.057	0.041
CoreHD	0.051	0.113
EGND	0.124	0.034
EI σ_1	0.057	0.115
EI σ_2	0.159	0.035
NES	0.147	0.088
NESR	0.067	0.031
NEM	0.146	0.127
NEMR	0.067	0.024
NEL	0.184	0.091
NELR	0.073	0.018
VE	0.121	0.070
VER	0.064	0.054

Table S4. **Ablation study results on synthetic and real-world network datasets.** This table evaluates the impact of three key components on the prediction error MAE $\downarrow$: (1) JK type, comparing the LSTM-based layer aggregation strategy (ours) with using only the last-layer output (Last); (2) Residual, comparing the model with and without residual connections; and (3) Virtual node, comparing the model with and without the virtual global node. “✓” indicates that the component is enabled, while “×” indicates that it is removed. All results are averaged over independent runs and reported as mean $\pm$ standard deviation.

Dataset	JK type	Residual	Virtual node	MAE
Small	LSTM	✓	✓	0.011 ± 0.0001
	Last	✓	✓	0.011 ± 0.0007
	LSTM	×	✓	0.010 ± 0.0001
	LSTM	✓	×	0.013 ± 0.0009
Medium	LSTM	✓	✓	0.008 ± 0.0005
	Last	✓	✓	0.007 ± 0.0013
	LSTM	×	✓	0.006 ± 0.0011
	LSTM	✓	×	0.008 ± 0.0003
Large	LSTM	✓	✓	0.005 ± 0.0007
	Last	✓	✓	0.008 ± 0.0029
	LSTM	×	✓	0.004 ± 0.0005
	LSTM	✓	×	0.007 ± 0.0003
Extra-large	LSTM	✓	✓	0.005 ± 0.0009
	Last	✓	✓	0.005 ± 0.0013
	LSTM	×	✓	0.004 ± 0.0001
	LSTM	✓	×	0.006 ± 0.0015
REDDIT	LSTM	✓	✓	0.008 ± 0.0008
	Last	✓	✓	0.021 ± 0.0024
	LSTM	×	✓	0.006 ± 0.0007
	LSTM	✓	×	0.005 ± 0.0002

Table S5. **Ablation study results on attack scenario datasets.** This table evaluates the impact of the same three components on MAE $\downarrow$, M $\downarrow$, S $\downarrow$. The component configurations follow Table S4, where “ $\checkmark/\times$ ” denote enabling/removing a component. Note: the values of M and S are scaled by a factor of $\times 10^2$ in all tables and figures for readability. All results are reported as mean $\pm$ standard deviation over independent runs. **Blue text** indicates the best result for each dataset, and **gray shading** indicates the second-best.

Dataset	JK type	Residual	Virtual node	MAE	$M (\times 10^2)$	$S (\times 10^2)$
BA100	LSTM	$\checkmark$	$\checkmark$	0.019 ± 0.0016	0.0	0.7 ± 1.0
	Last	$\checkmark$	$\checkmark$	0.024 ± 0.0038	3.0 ± 0.8	6.7 ± 2.0
	LSTM	$\times$	$\checkmark$	0.396 ± 0.2109	4.8 ± 3.4	20.6 ± 9.2
	LSTM	$\checkmark$	$\times$	0.022 ± 0.0028	0.1 ± 0.1	2.6 ± 2.4
LFR100	LSTM	$\checkmark$	$\checkmark$	0.014 ± 0.0005	0.0	0.5 ± 0.1
	Last	$\checkmark$	$\checkmark$	0.018 ± 0.0017	0.2 ± 0.4	1.4 ± 0.5
	LSTM	$\times$	$\checkmark$	0.398 ± 0.2025	1.2 ± 2.2	18.0 ± 12.1
	LSTM	$\checkmark$	$\times$	0.015 ± 0.0019	0.0	2.3 ± 2.1
ER2000	LSTM	$\checkmark$	$\checkmark$	0.015 ± 0.0033	0.0	2.0 ± 1.3
	Last	$\checkmark$	$\checkmark$	0.020 ± 0.0060	0.1 ± 0.2	2.6 ± 0.9
	LSTM	$\times$	$\checkmark$	0.212 ± 0.0341	1.7 ± 1.7	4.1 ± 3.0
	LSTM	$\checkmark$	$\times$	0.024 ± 0.0051	0.0	3.1 ± 0.1
WS2000	LSTM	$\checkmark$	$\checkmark$	0.012 ± 0.0020	0.0	0.3 ± 0.1
	Last	$\checkmark$	$\checkmark$	0.012 ± 0.0016	0.0	0.5 ± 0.6
	LSTM	$\times$	$\checkmark$	0.378 ± 0.1690	1.6 ± 0.7	5.8 ± 4.0
	LSTM	$\checkmark$	$\times$	0.016 ± 0.0017	0.0	3.7 ± 2.3
Transport	LSTM	$\checkmark$	$\checkmark$	0.010 ± 0.0006	0.0	0.5 ± 0.2
	Last	$\checkmark$	$\checkmark$	0.009 ± 0.0010	0.0 ± 0.1	0.9 ± 1.0
	LSTM	$\times$	$\checkmark$	0.135 ± 0.0577	2.1 ± 1.8	25.9 ± 5.5
	LSTM	$\checkmark$	$\times$	0.010 ± 0.0004	0.0	0.4 ± 0.3
Power	LSTM	$\checkmark$	$\checkmark$	0.029 ± 0.0106	1.6 ± 0.9	2.8 ± 1.5
	Last	$\checkmark$	$\checkmark$	0.056 ± 0.0155	3.3 ± 0.9	4.1 ± 0.9
	LSTM	$\times$	$\checkmark$	0.225 ± 0.0660	4.8 ± 3.0	4.8 ± 4.6
	LSTM	$\checkmark$	$\times$	0.025 ± 0.0032	2.5 ± 1.6	2.0 ± 1.0

XII. SUPPLEMENTARY FIGURES

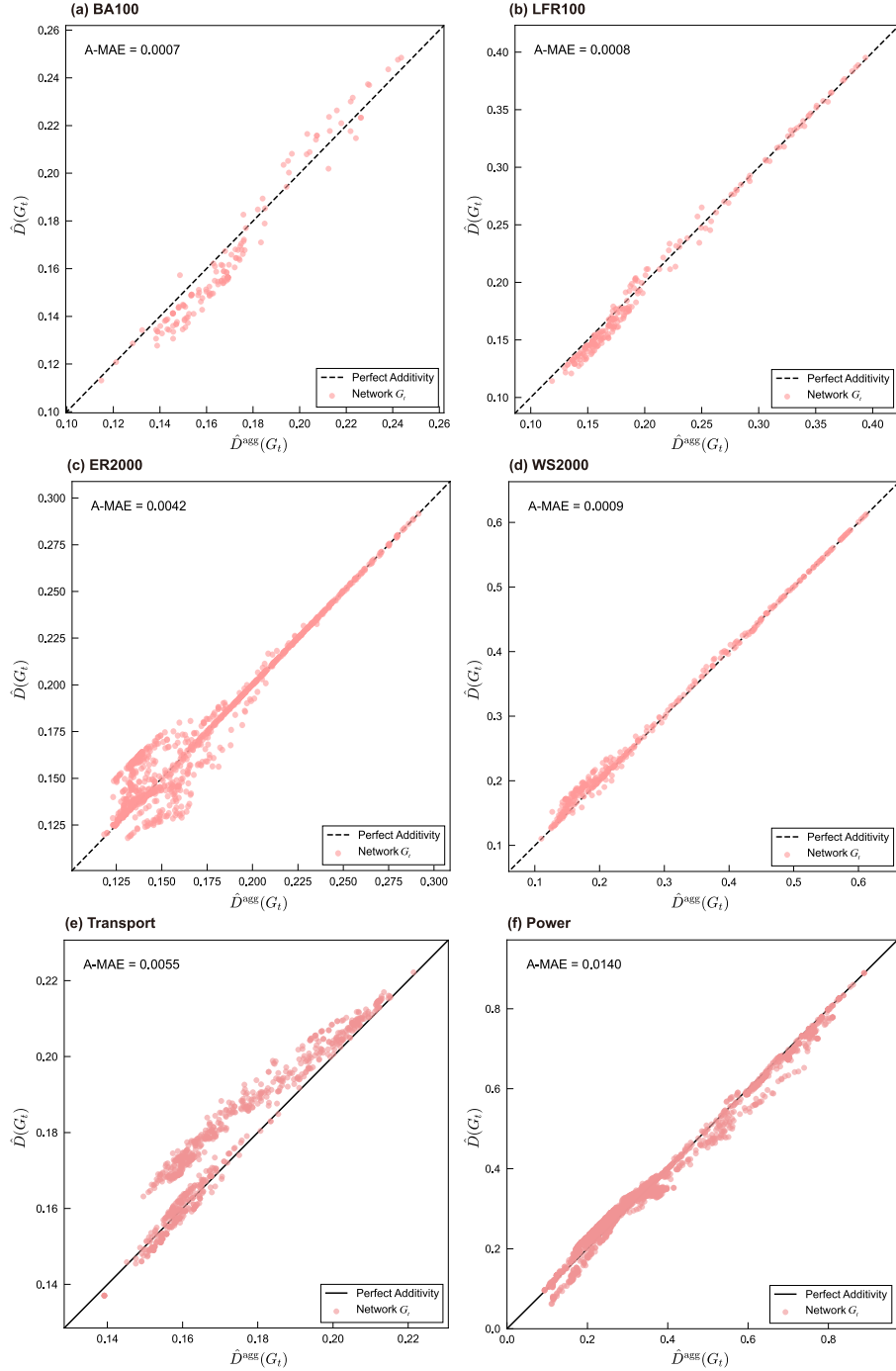


Figure S1. **Verification of scale-transformed linear additivity.** Scatter plots compare the direct global prediction $\hat{D}(G_t)$ (y-axis) with the component-wise aggregated prediction $\hat{D}^{\text{agg}}(G_t)$ (x-axis) produced by TCR-GIN on four synthetic networks under attack-induced fragmentation. Test samples are snapshots where the network splits into multiple disconnected components ($k > 1$), collected across 30 distinct attack strategies. Each red point denotes one snapshot, and the dashed diagonal line ($y = x$) indicates perfect additivity.

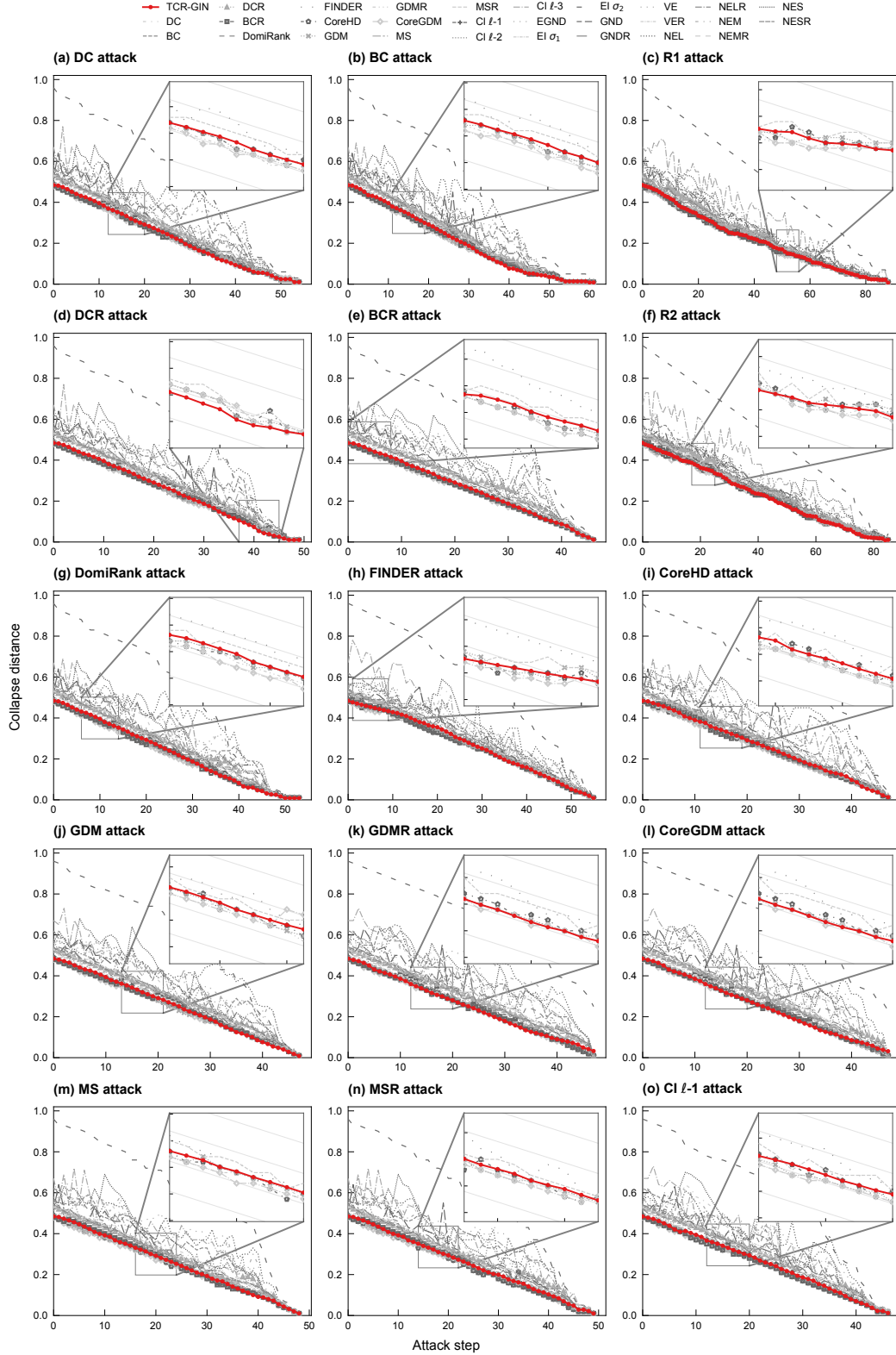


Figure S2. **Online collapse-distance trajectories of the BA100 network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S2 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified view. The

insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM, CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

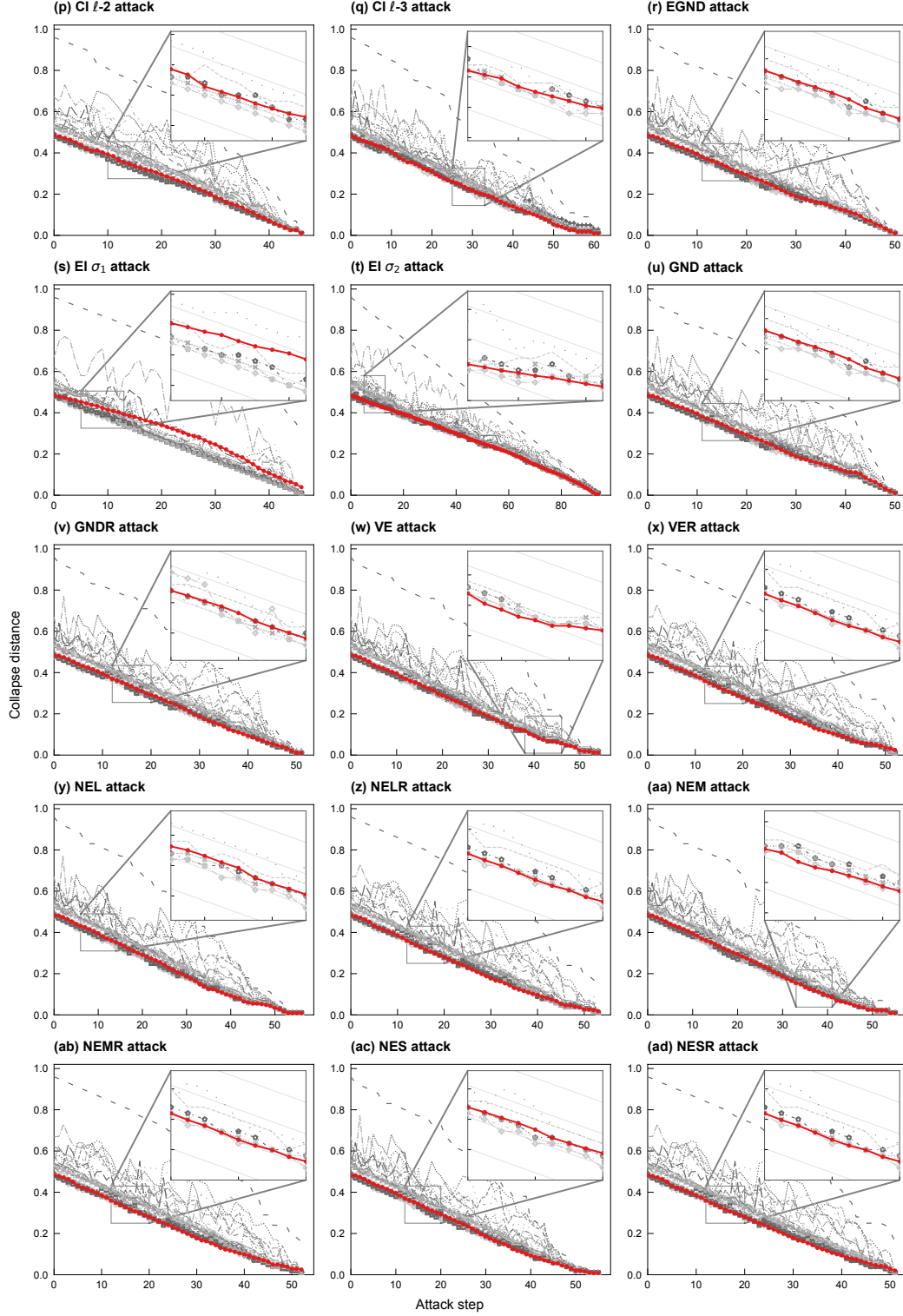


Figure S2 (continued).

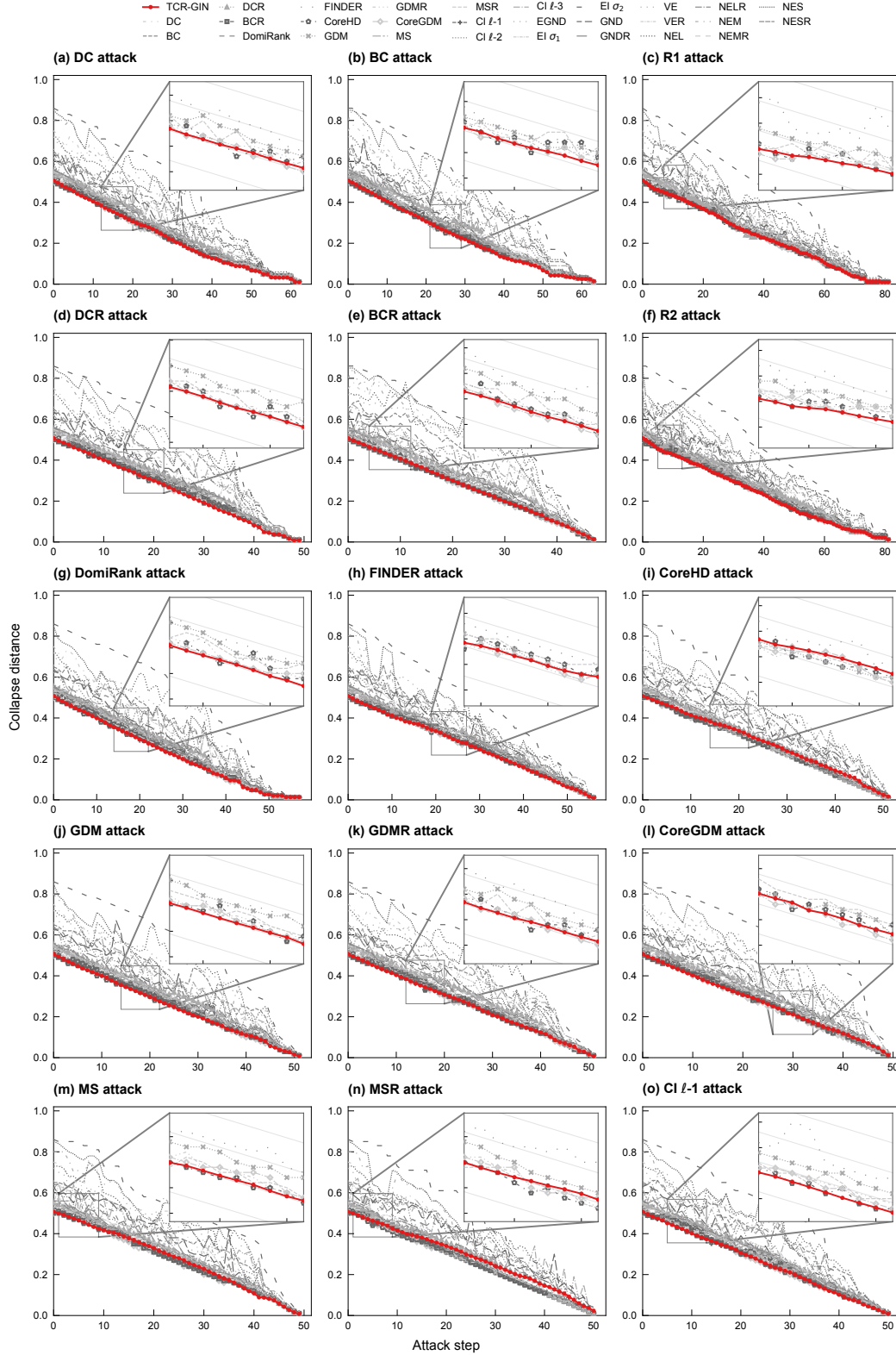


Figure S3. **Online collapse-distance trajectories of the LFR100 network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S3 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified

view. The insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM, CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

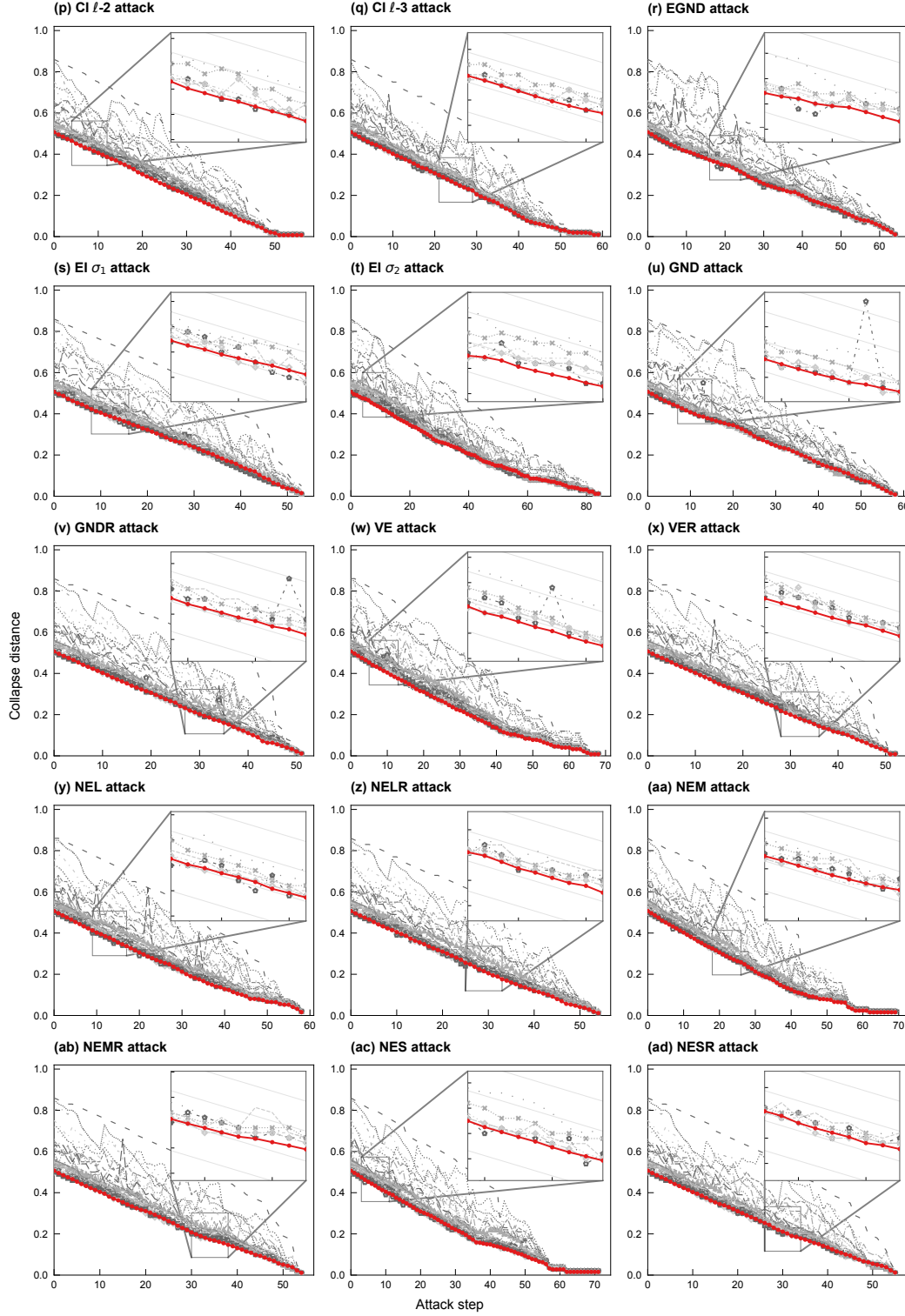


Figure S3 (continued).

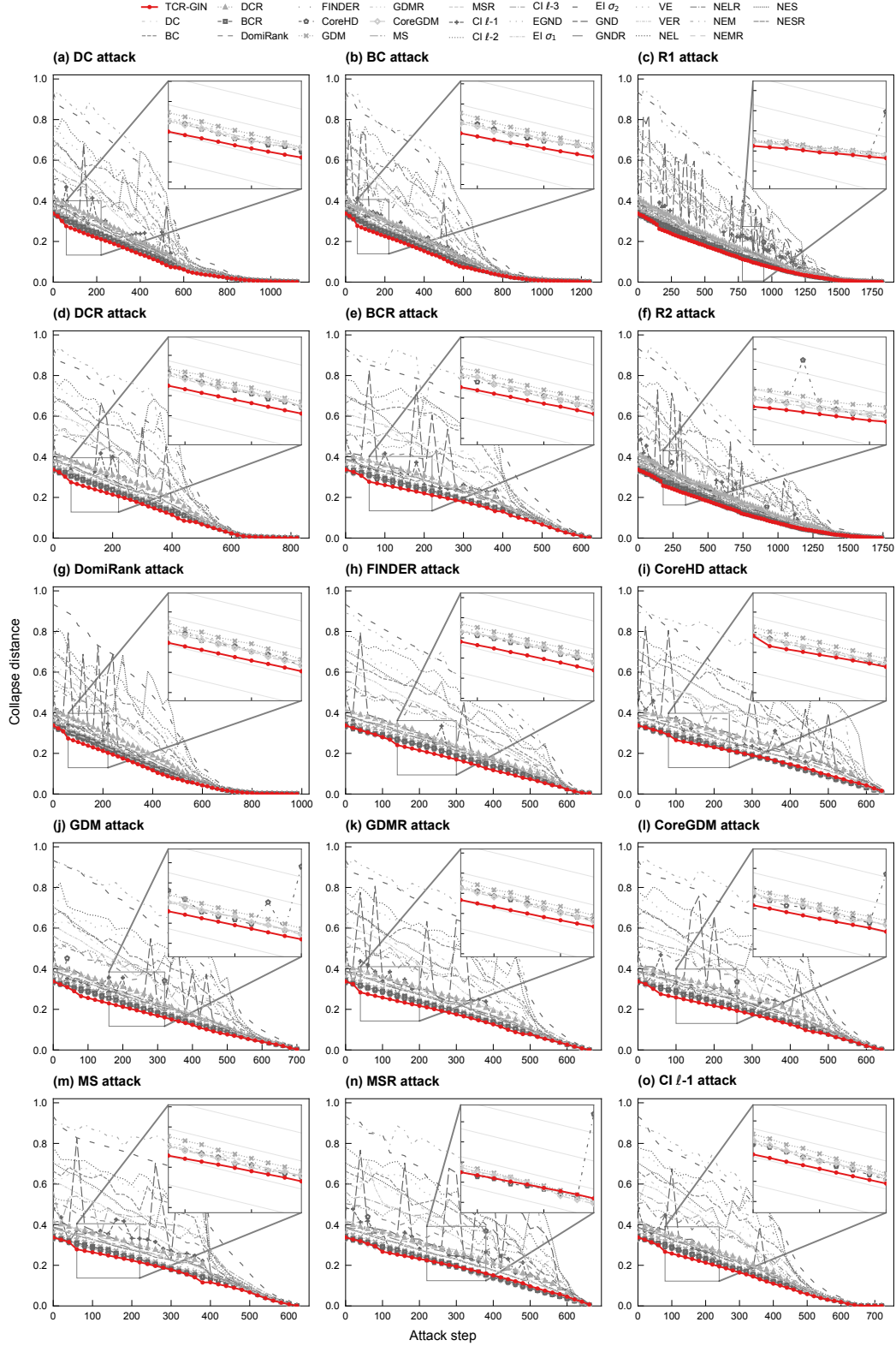


Figure S4. **Online collapse-distance trajectories of the ER2000 network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S4 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified

view. The insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM, CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

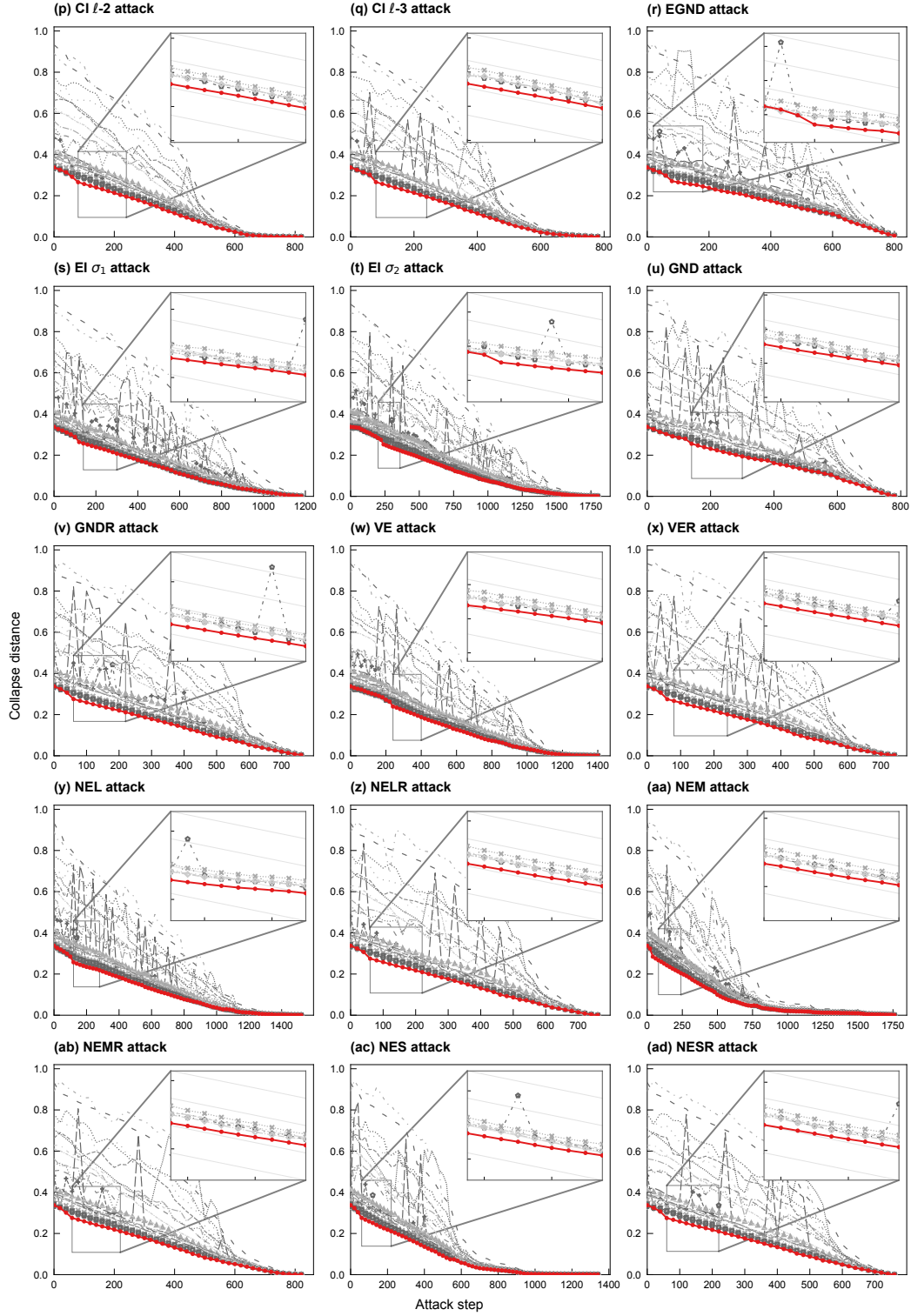


Figure S4 (continued).

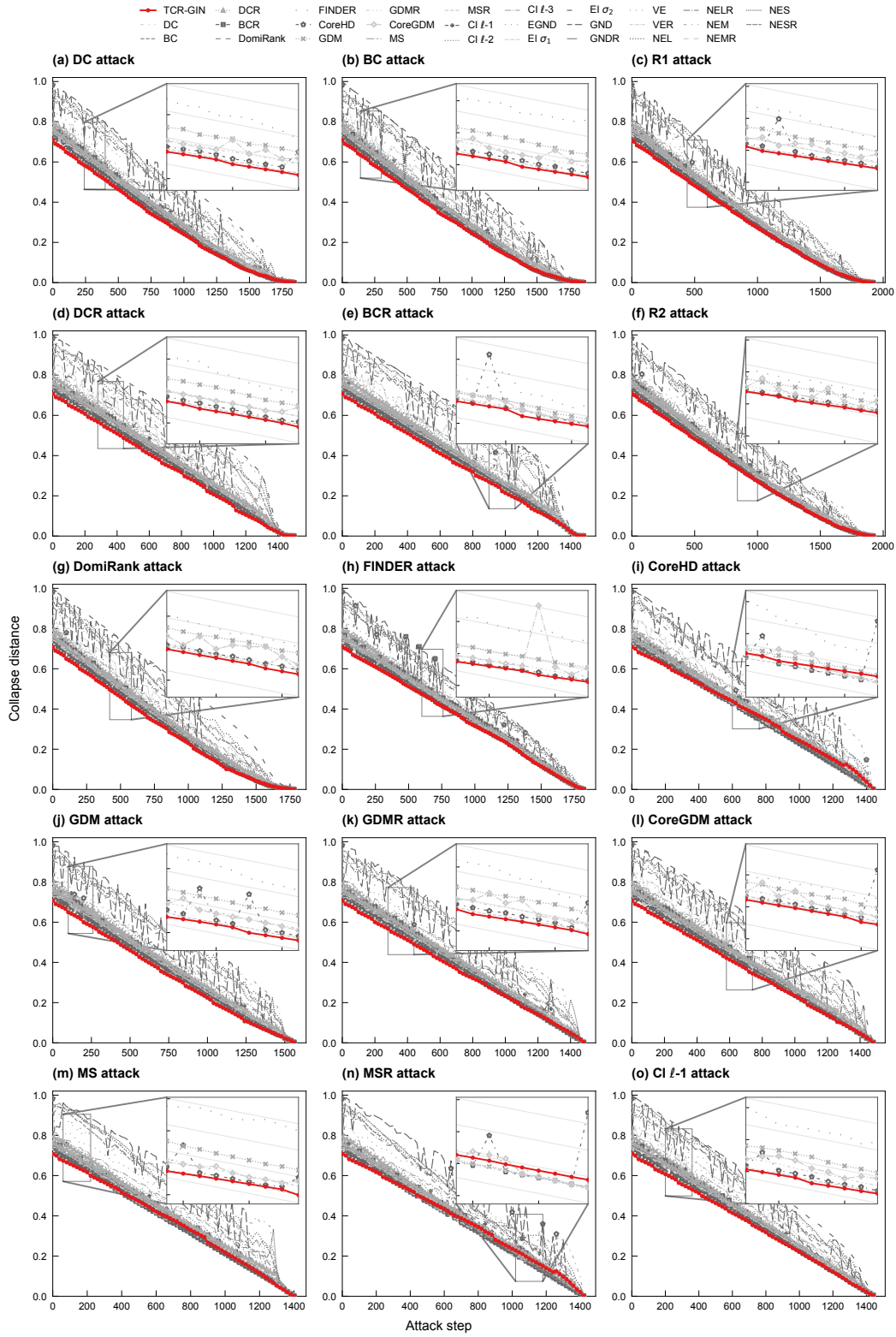


Figure S5. **Online collapse-distance trajectories of the WS2000 network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S5 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified view. The insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM,

CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

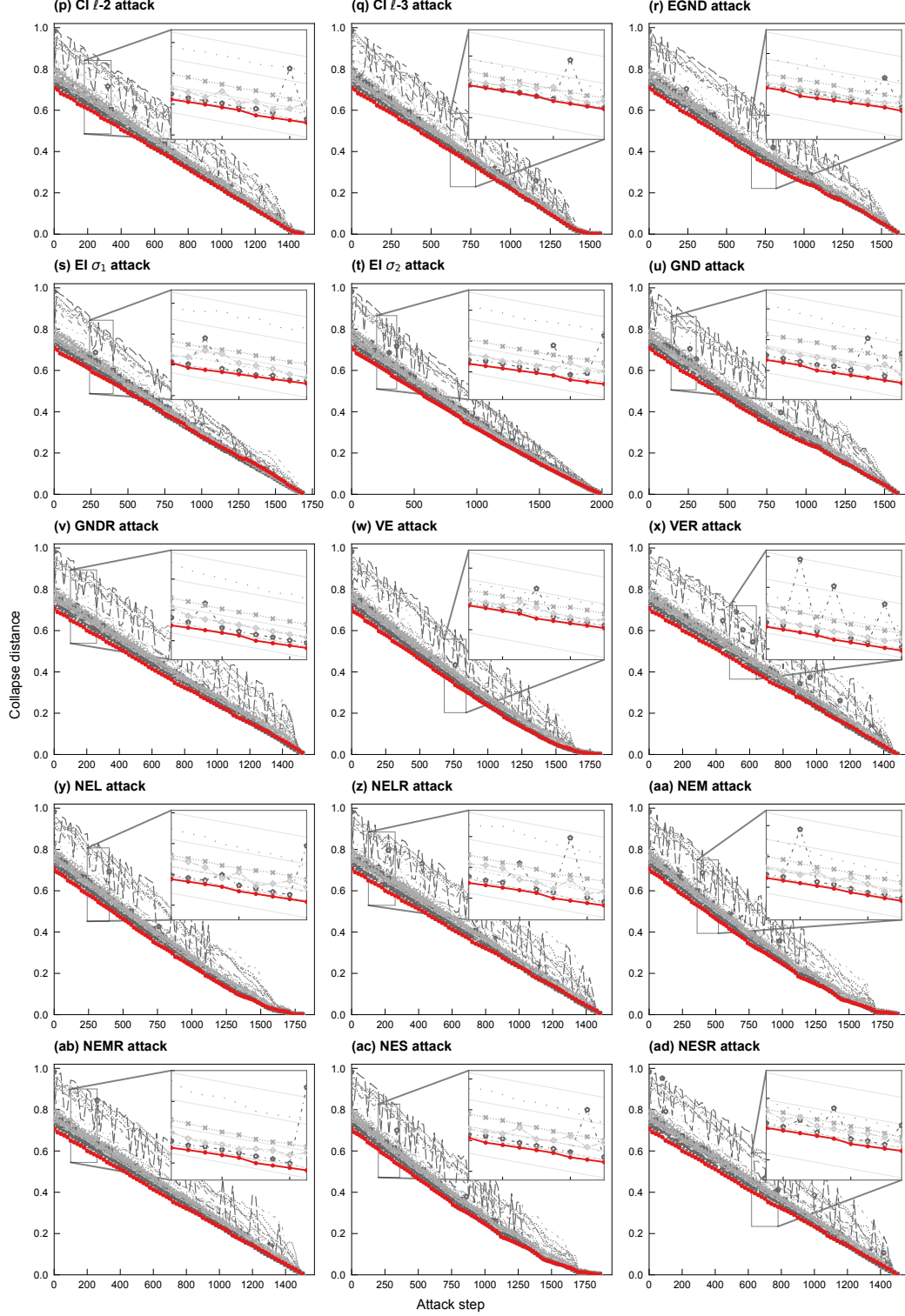


Figure S5 (continued).

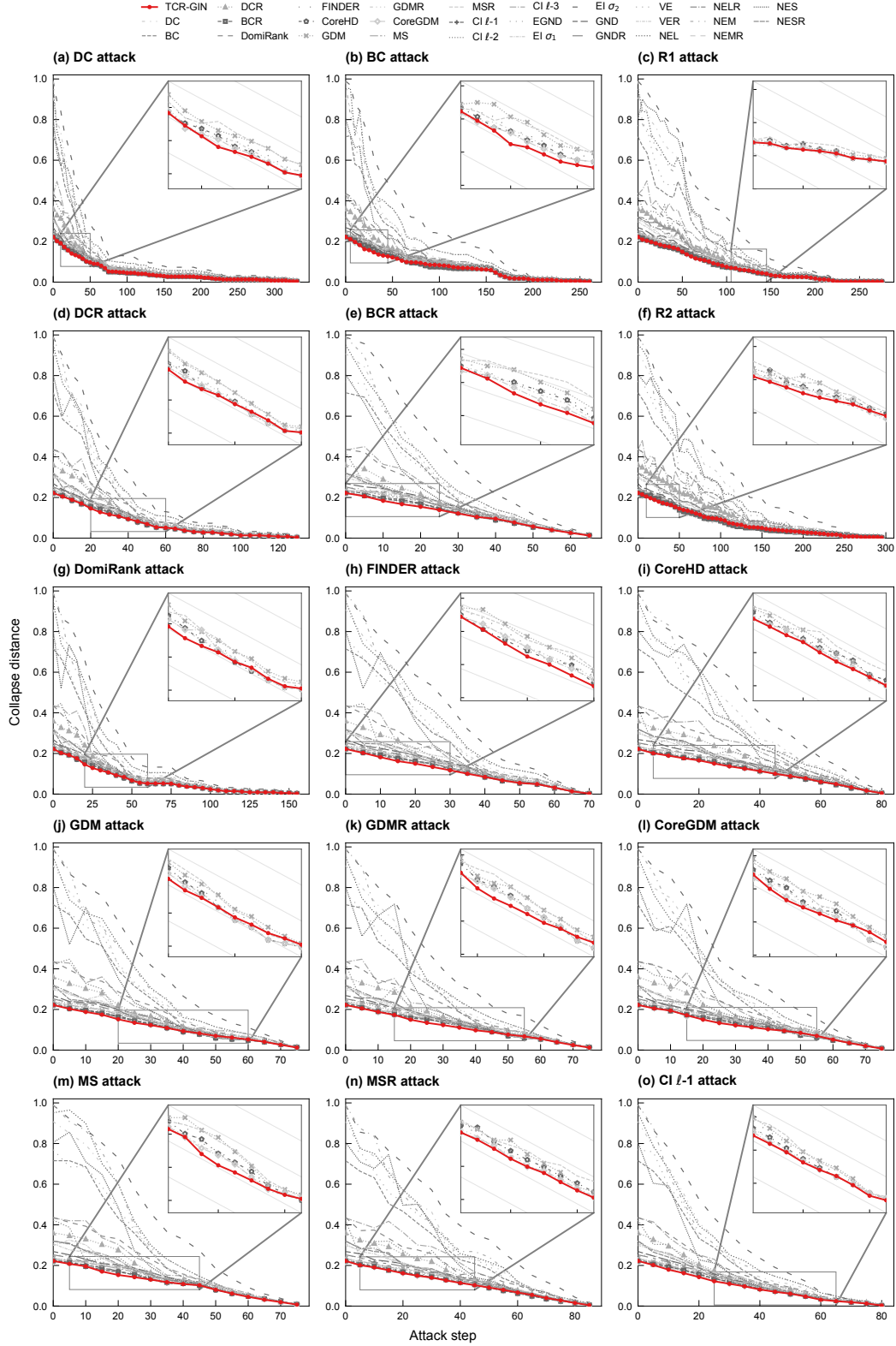


Figure S6. **Online collapse-distance trajectories of the Transport network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S6 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified

view. The insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM, CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

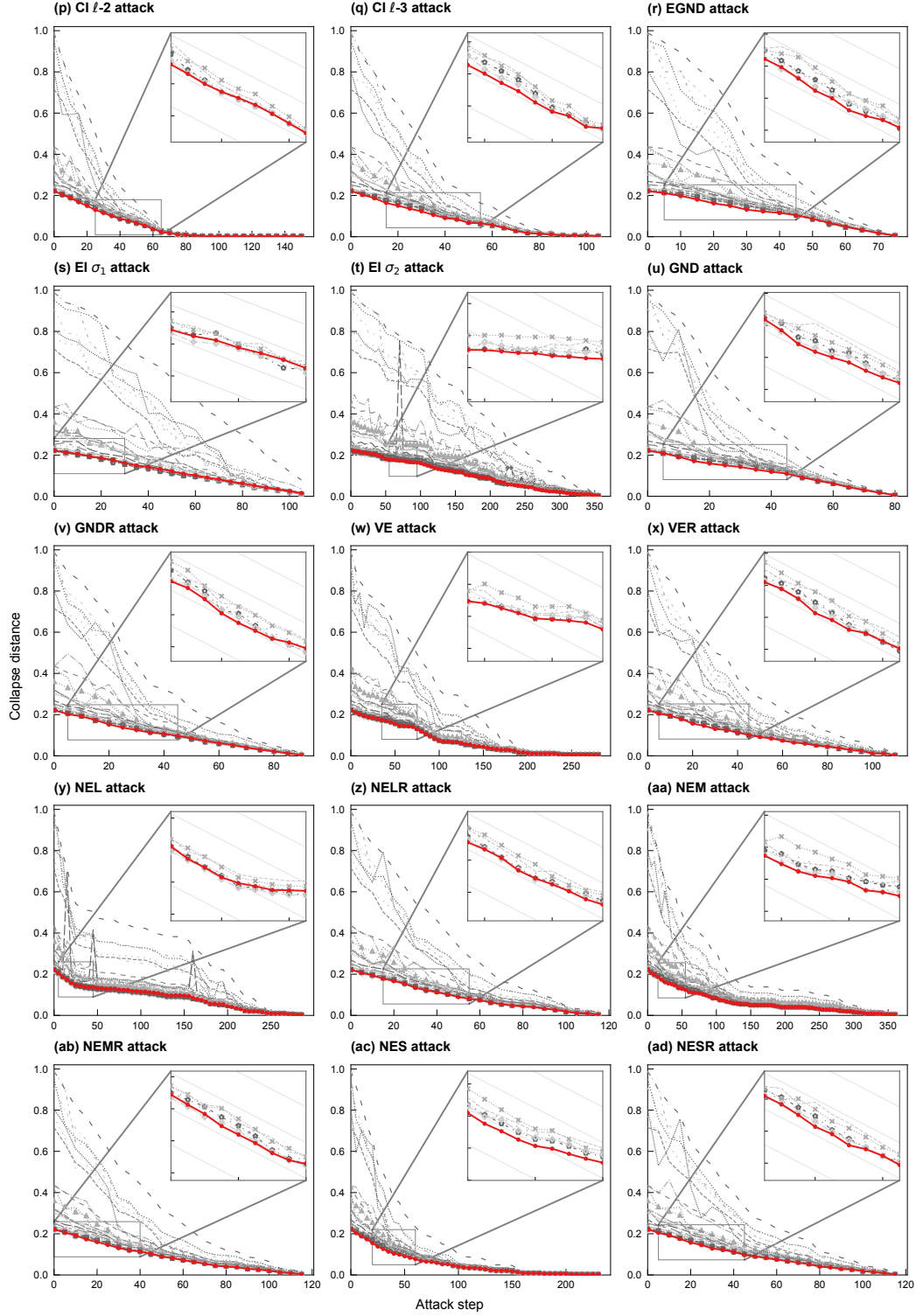


Figure S6 (continued).

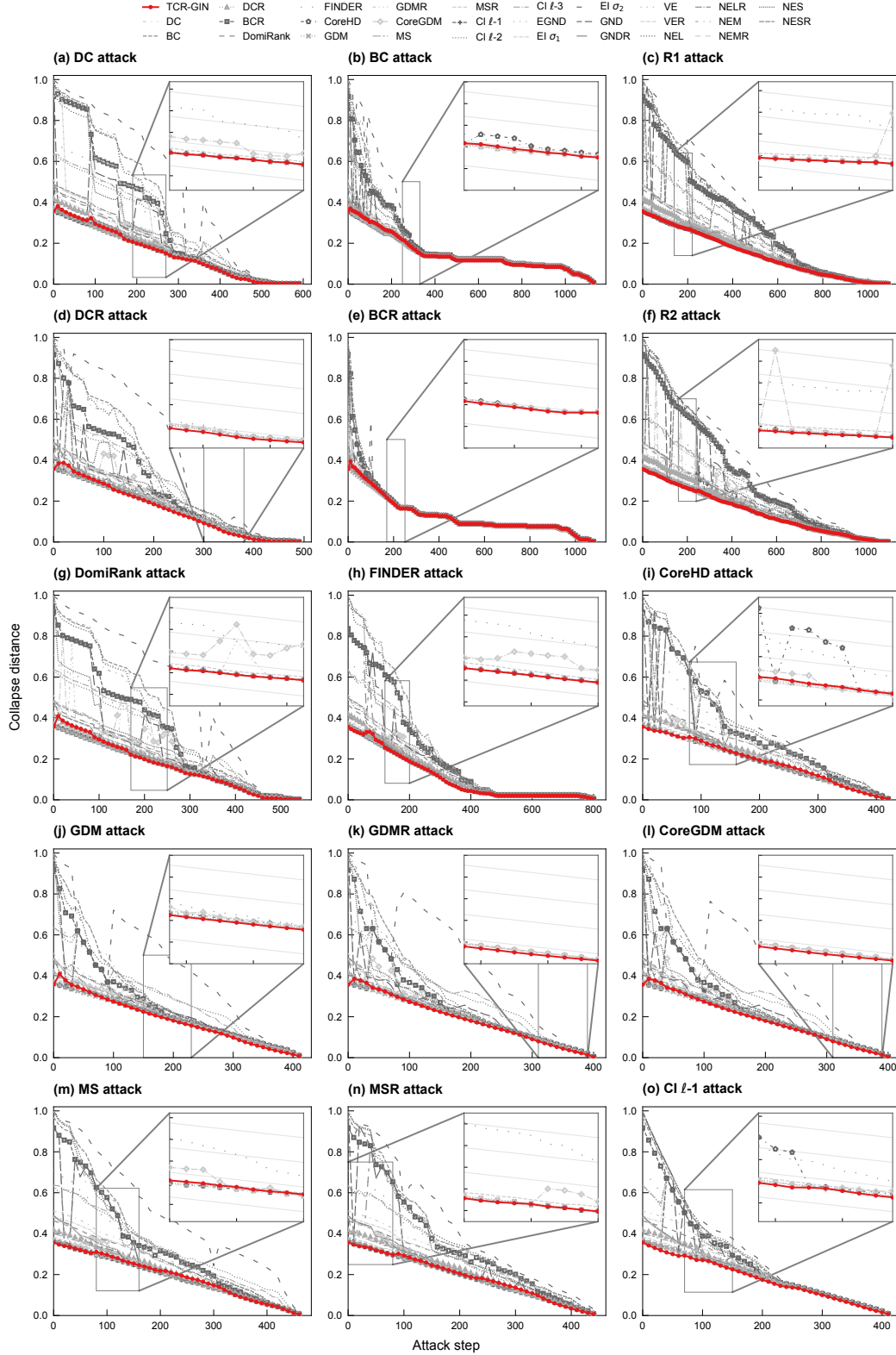


Figure S7. **Online collapse-distance trajectories of the Power network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Figure S7 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. Insets show a magnified view. The

insets include only TCR-GIN and six advanced baselines, including GDMR, GDM, CoreGDM, CoreHD, MSR, and FINDER. The reference grid in each inset has slope $-1/N_0$, where N_0 is the initial number of nodes used for sequence normalization, visualizing the theoretical upper bound of the per-step change in collapse distance under one-node removal.

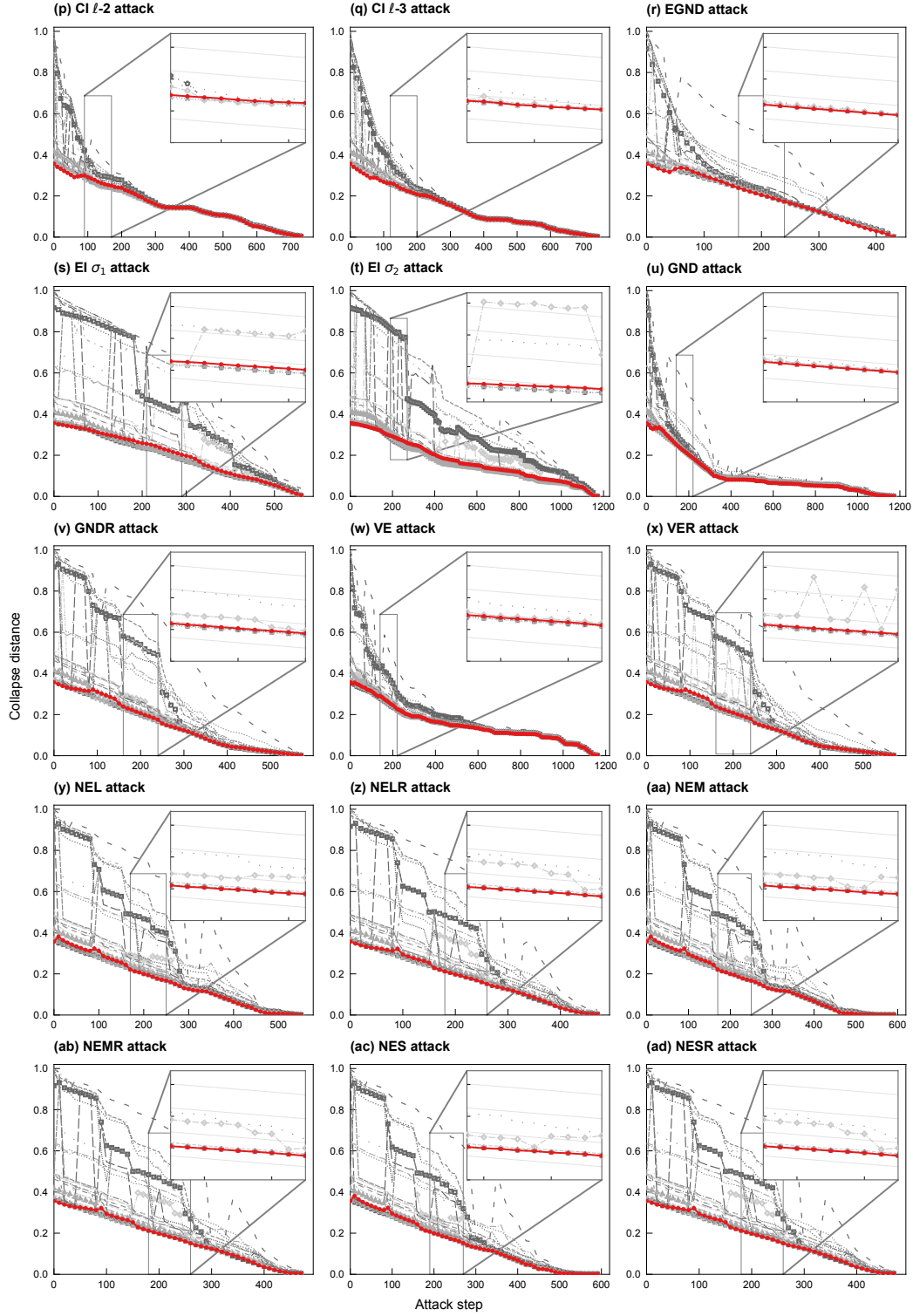


Figure S7 (continued).

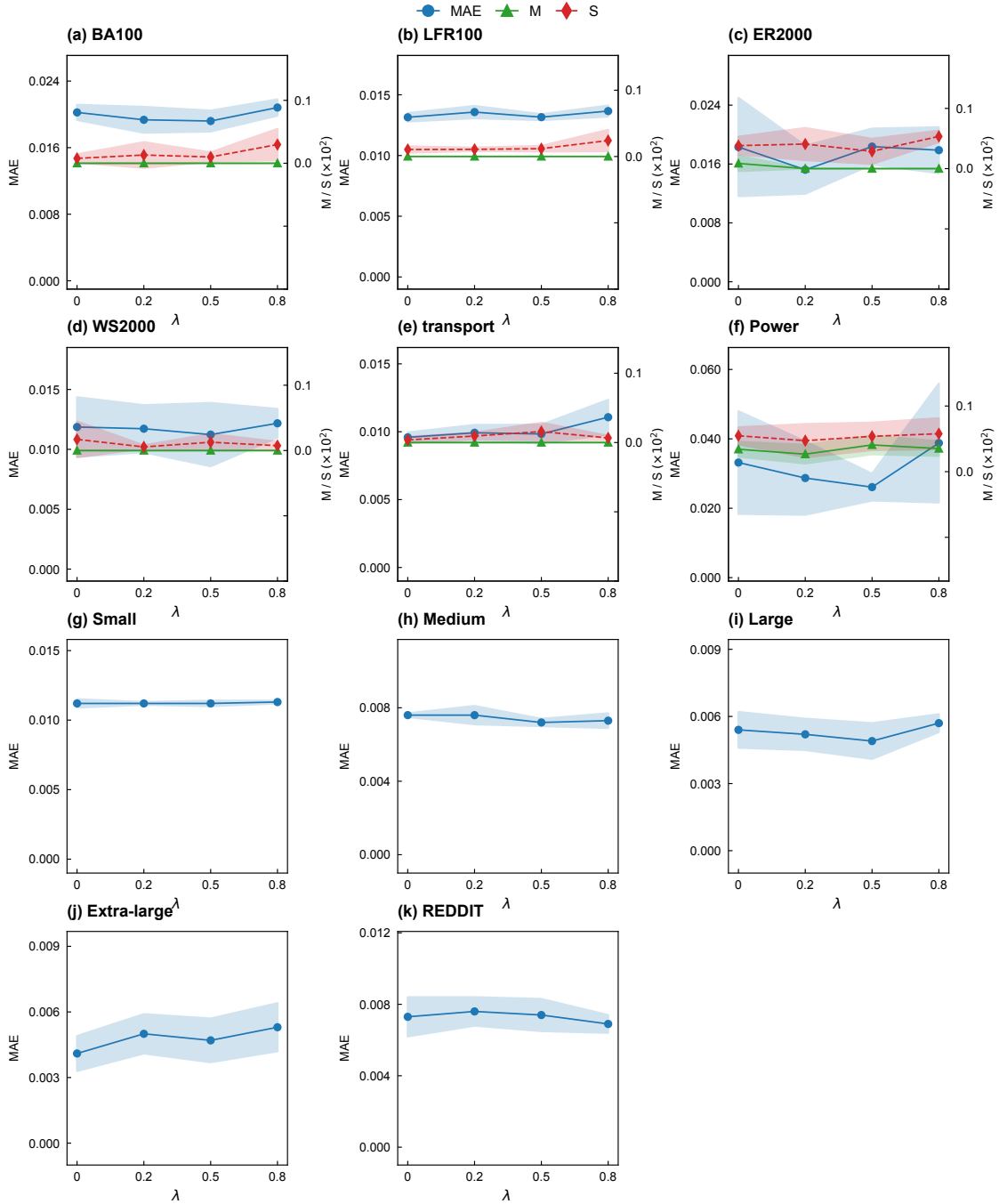


Figure S8. **Sensitivity analysis of the consistency regularization weight λ .** This figure illustrates the impact of varying $\lambda \in \{0, 0.2, 0.5, 0.8\}$ on model performance. (a)–(f) present the MAE (left axis), as well as Monotonicity (M) and Smoothness (S) violation frequencies (right axis; values scaled by 10^2) across attack scenarios. (g)–(k) show the MAE performance on static datasets. Note that $\lambda = 0$ corresponds to using the baseline supervised loss only, while increasing λ reinforces the model’s focus on consistency constraints.

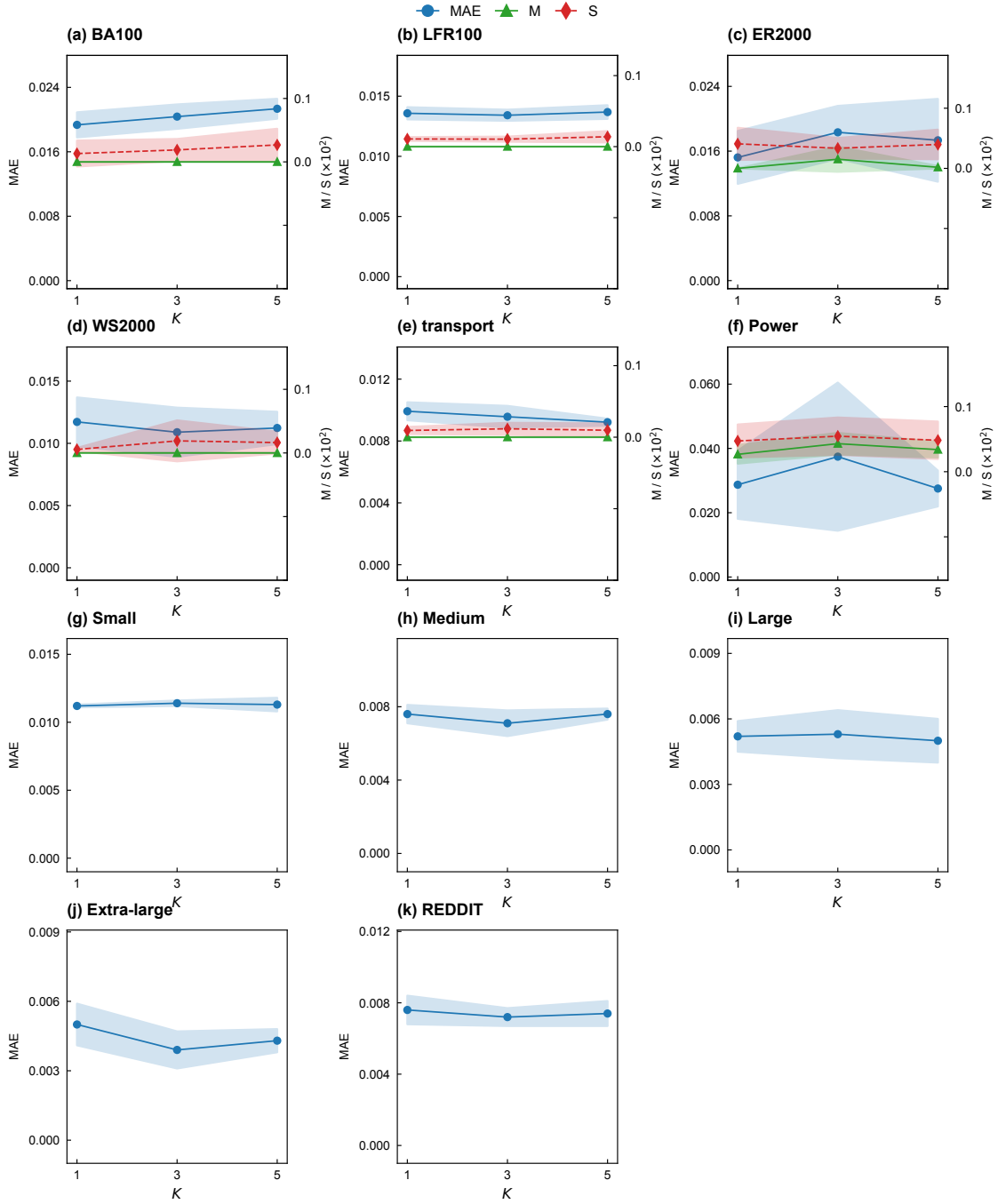


Figure S9. **Sensitivity analysis of the sampling size K .** This figure illustrates the impact of different sampling counts $K \in \{1, 3, 5\}$ on model performance. (a)–(f) Present the MAE (left axis), as well as Monotonicity (M) and Smoothness (S) violation frequencies (right axis; values scaled by 10^2) across attack scenarios. (g)–(k) Show the MAE performance on static datasets.

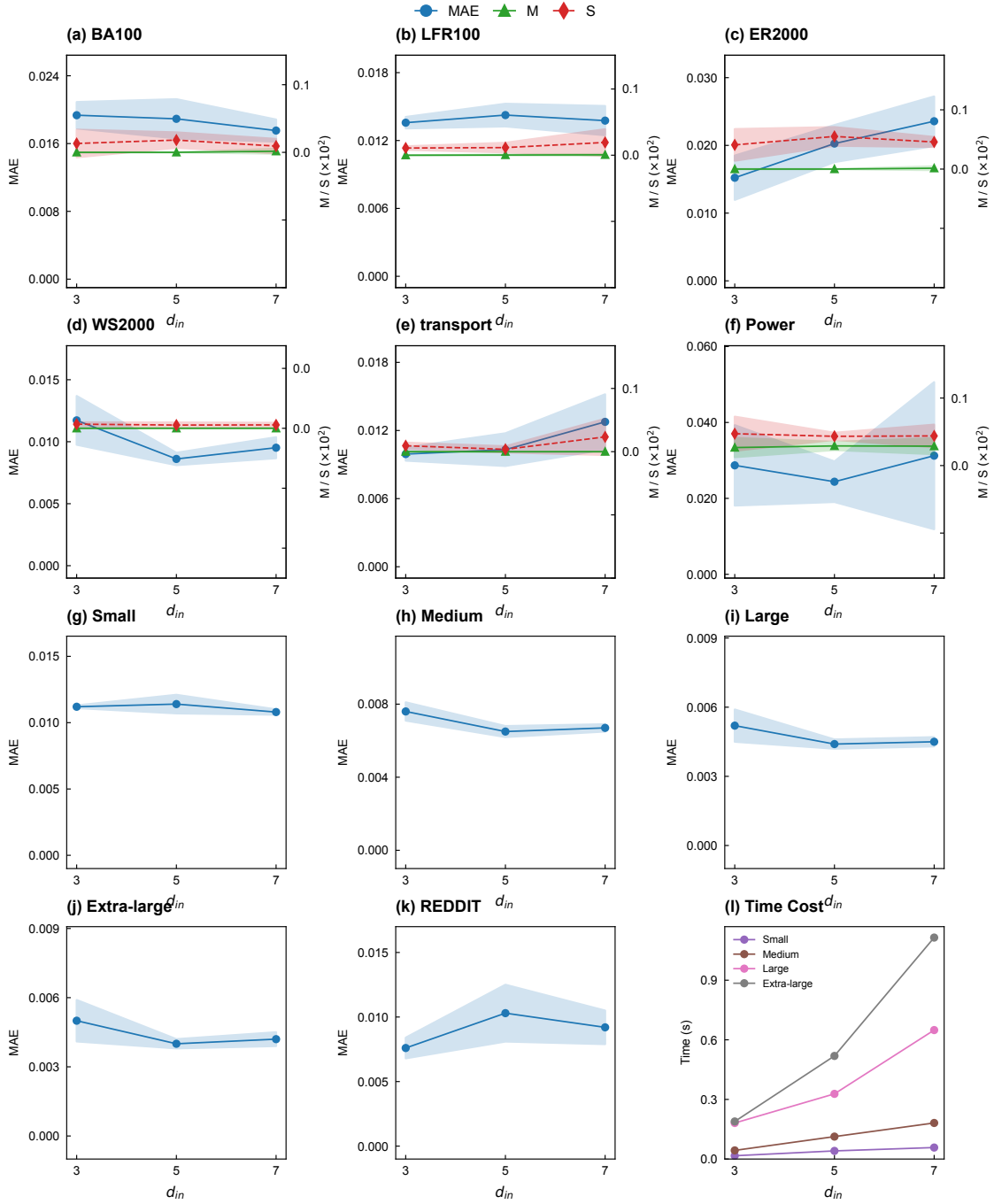


Figure S10. **Sensitivity analysis of the input feature dimension d_{in} .** This figure illustrates the impact of varying feature dimensions $d_{in} \in \{3, 5, 7\}$ on model performance. (a)–(f) Present the MAE (left axis), as well as Monotonicity (M) and Smoothness (S) violation frequencies (right axis; values scaled by 10^2) across attack scenarios. (g)–(k) Show the MAE performance on static datasets. (l) compares the inference efficiency on synthetic networks across different feature dimensions.

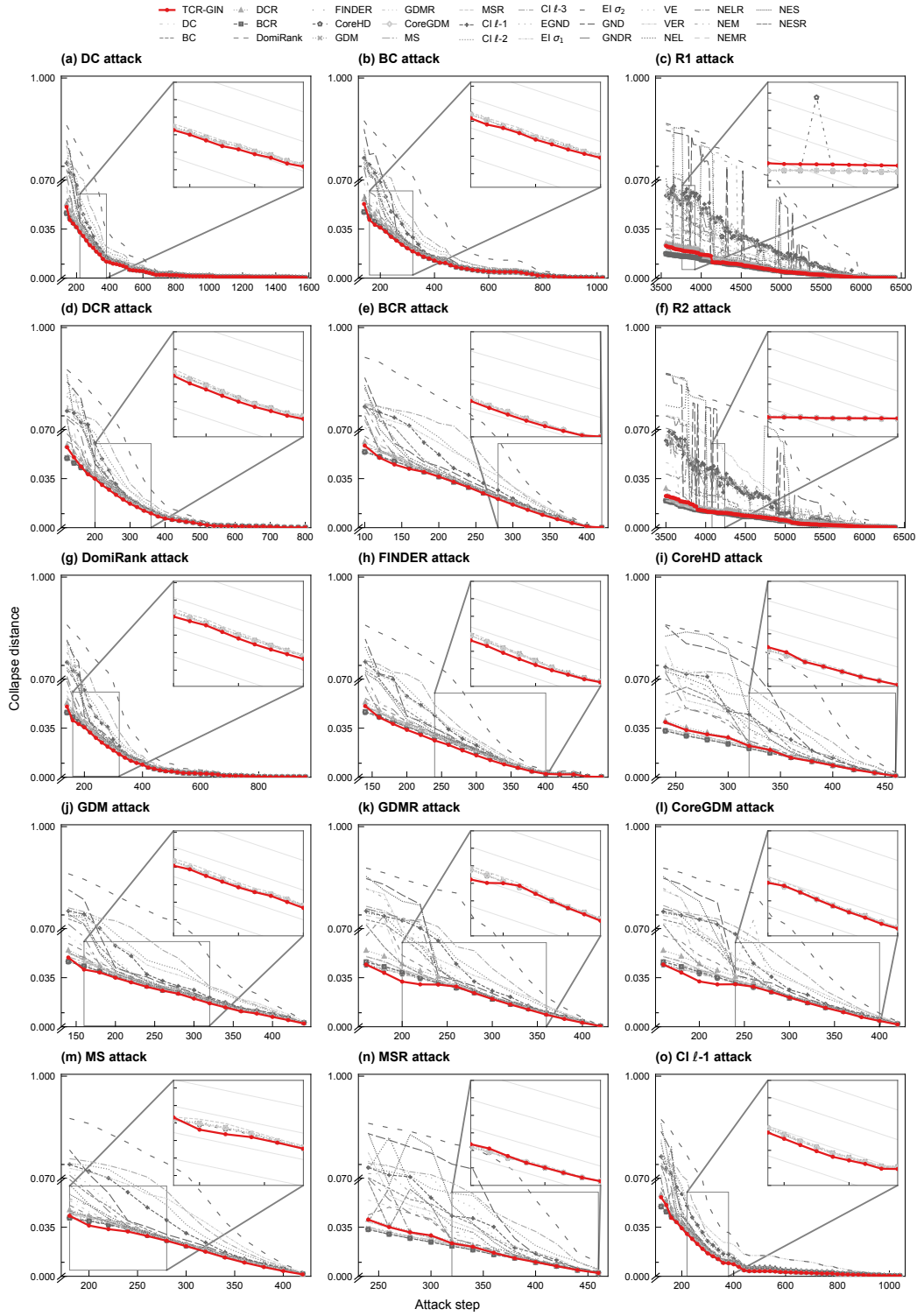


Figure S11. **Online collapse-distance trajectories of the Route network under sustained attacks.** (a–o) show 15 attack scenarios and (p–ad) are shown in Supplementary Figure S11 (continued). Red curves denote the TCR-GIN output, and grey curves denote the baseline algorithms. The Route network has 6,474 nodes. The model is designed for high-accuracy operation in the regime $|LCC_t| \leq 2000$. The first time point when $|LCC_t| \leq 2000$ is used as the activation point, and outputs and statistics are reported

only for the subsequent interval. All methods are compared on the same interval. The x-axis starts at different points across panels because each attack strategy reaches the $|LCC_t| \leq 2000$ regime at a different time.

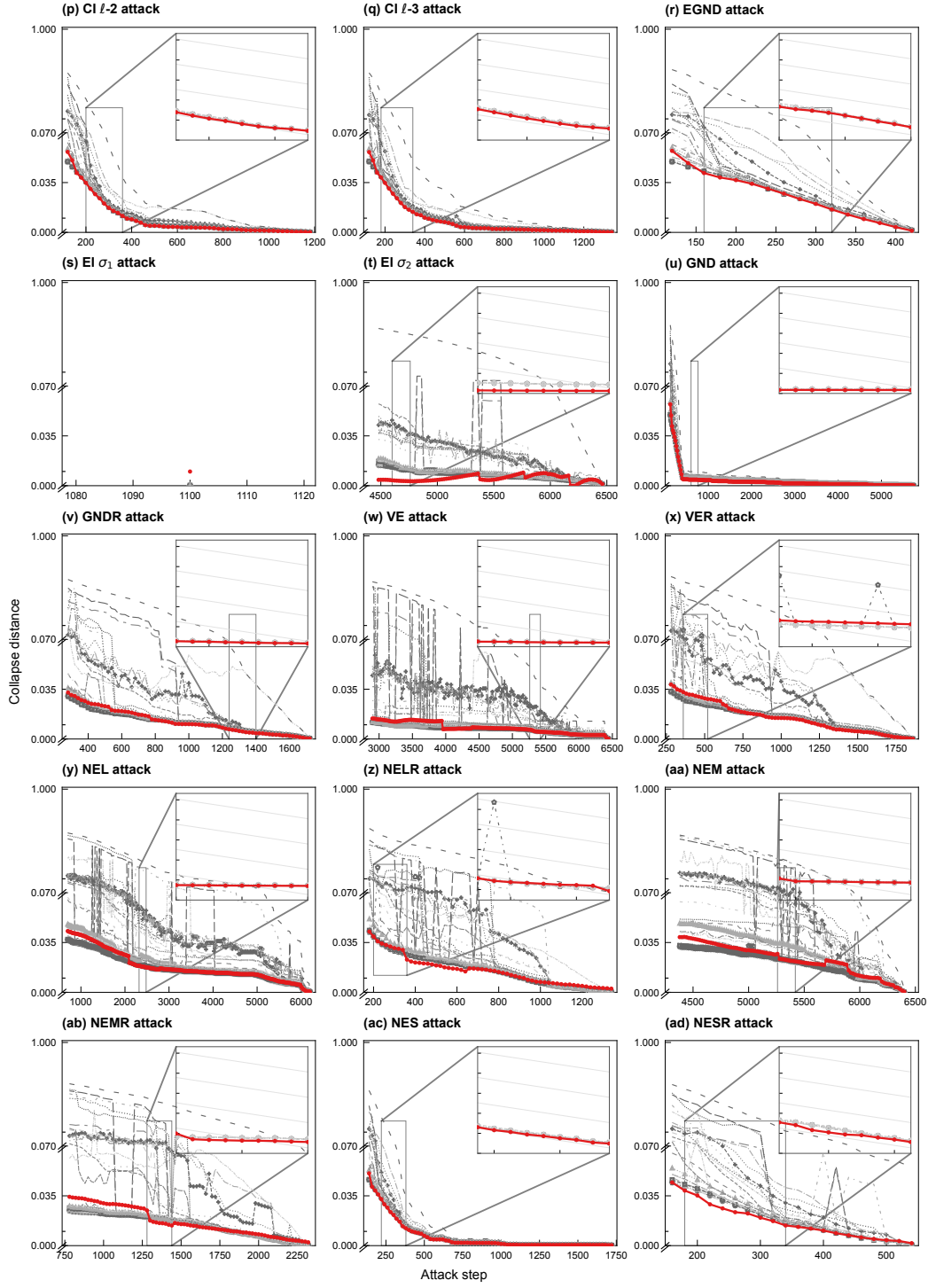


Figure S11 (continued).

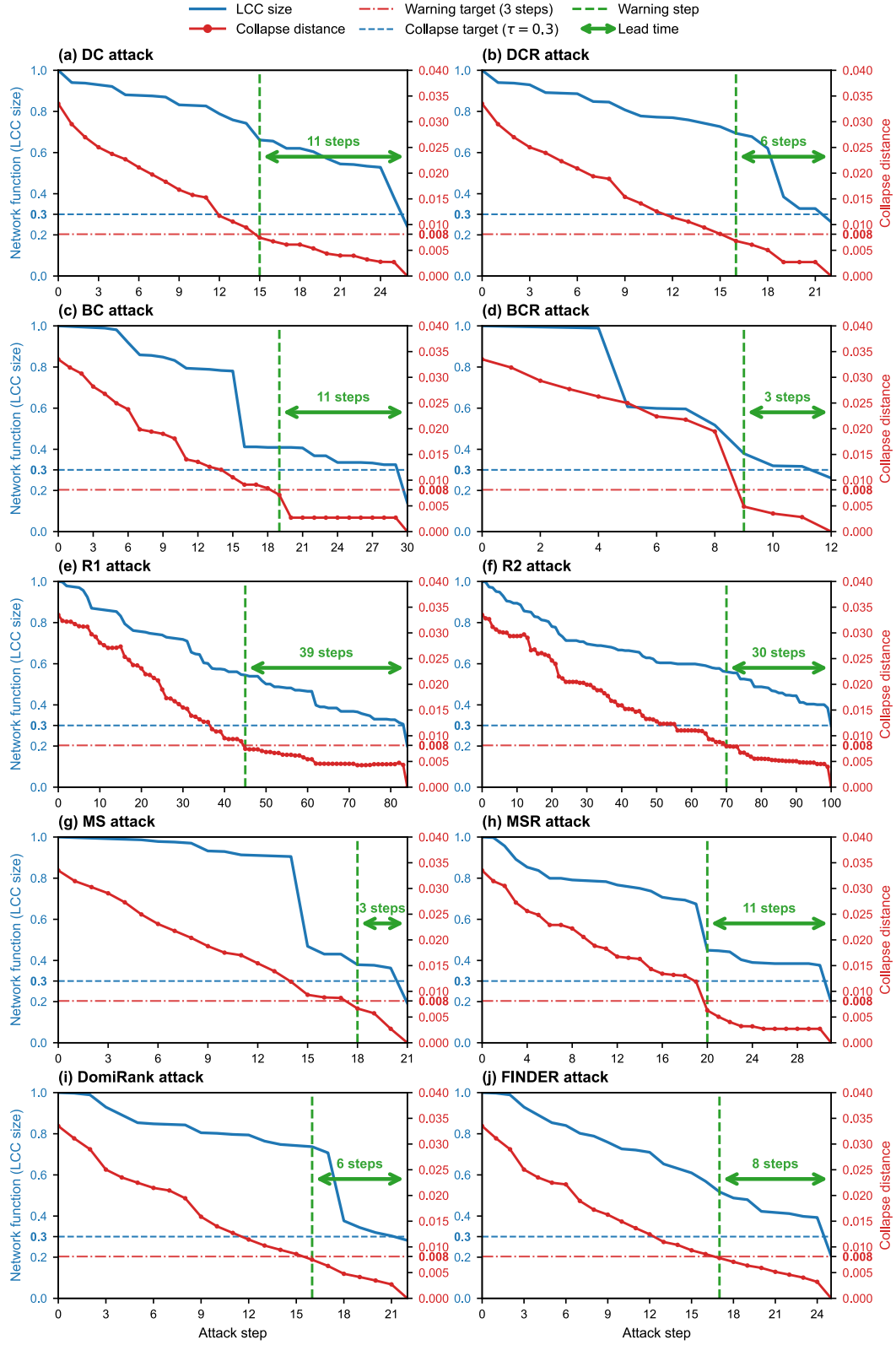


Figure S12. Early-warning signal of the Transport network under 30 attack scenarios at $\tau = 0.3$.

Plot elements are the same as in Fig. 6.

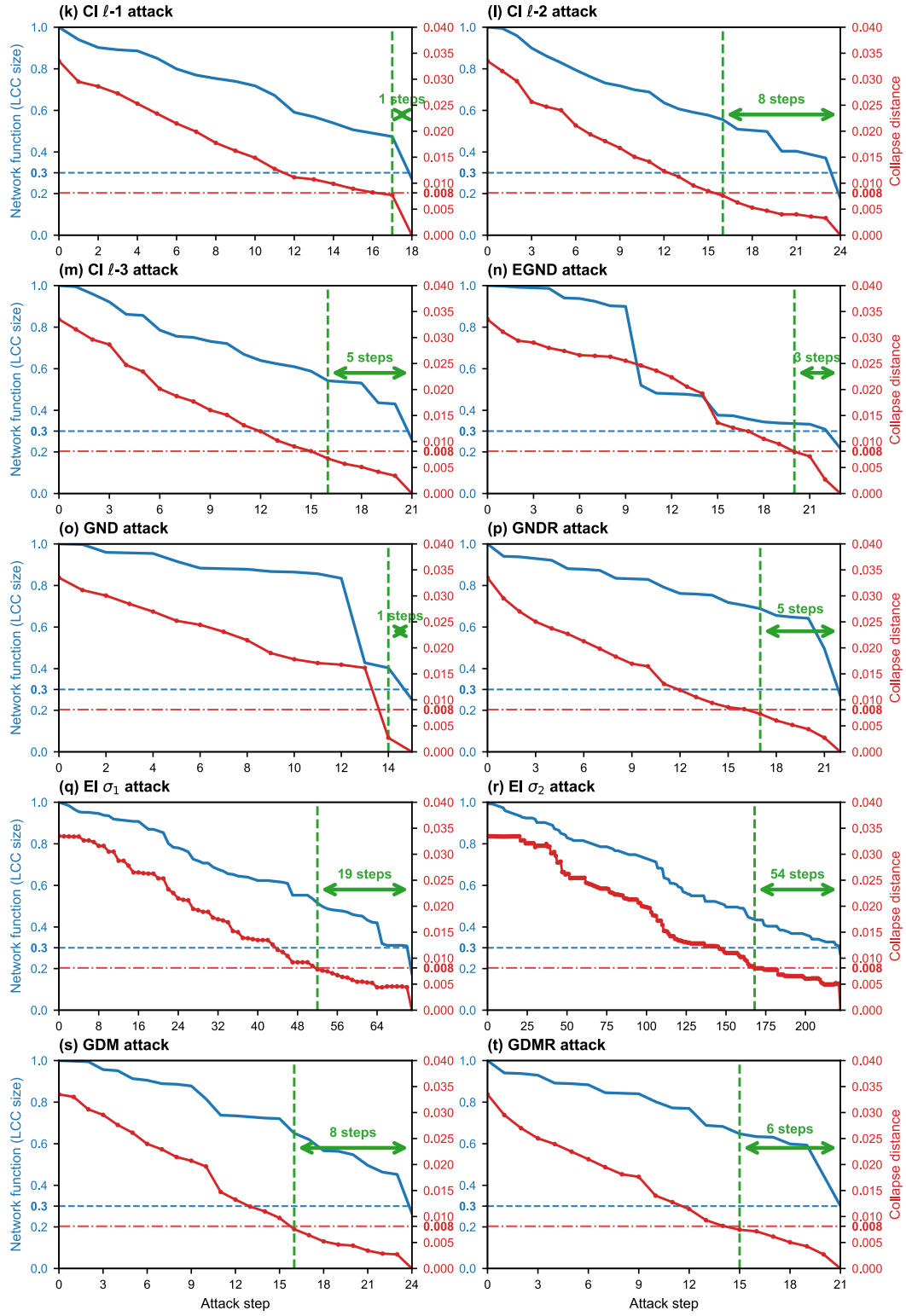


Figure S12 (continued).

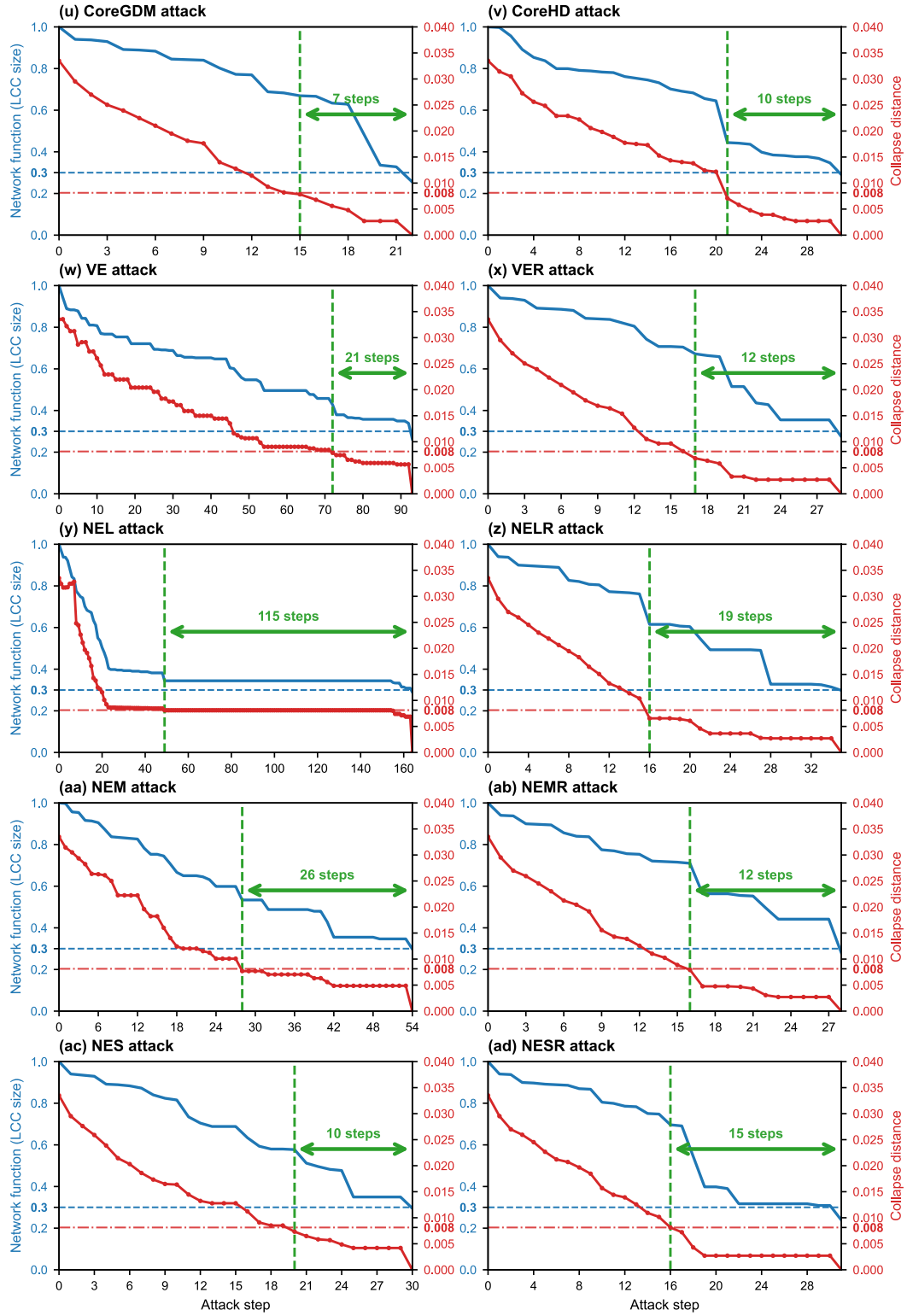


Figure S12 (continued).

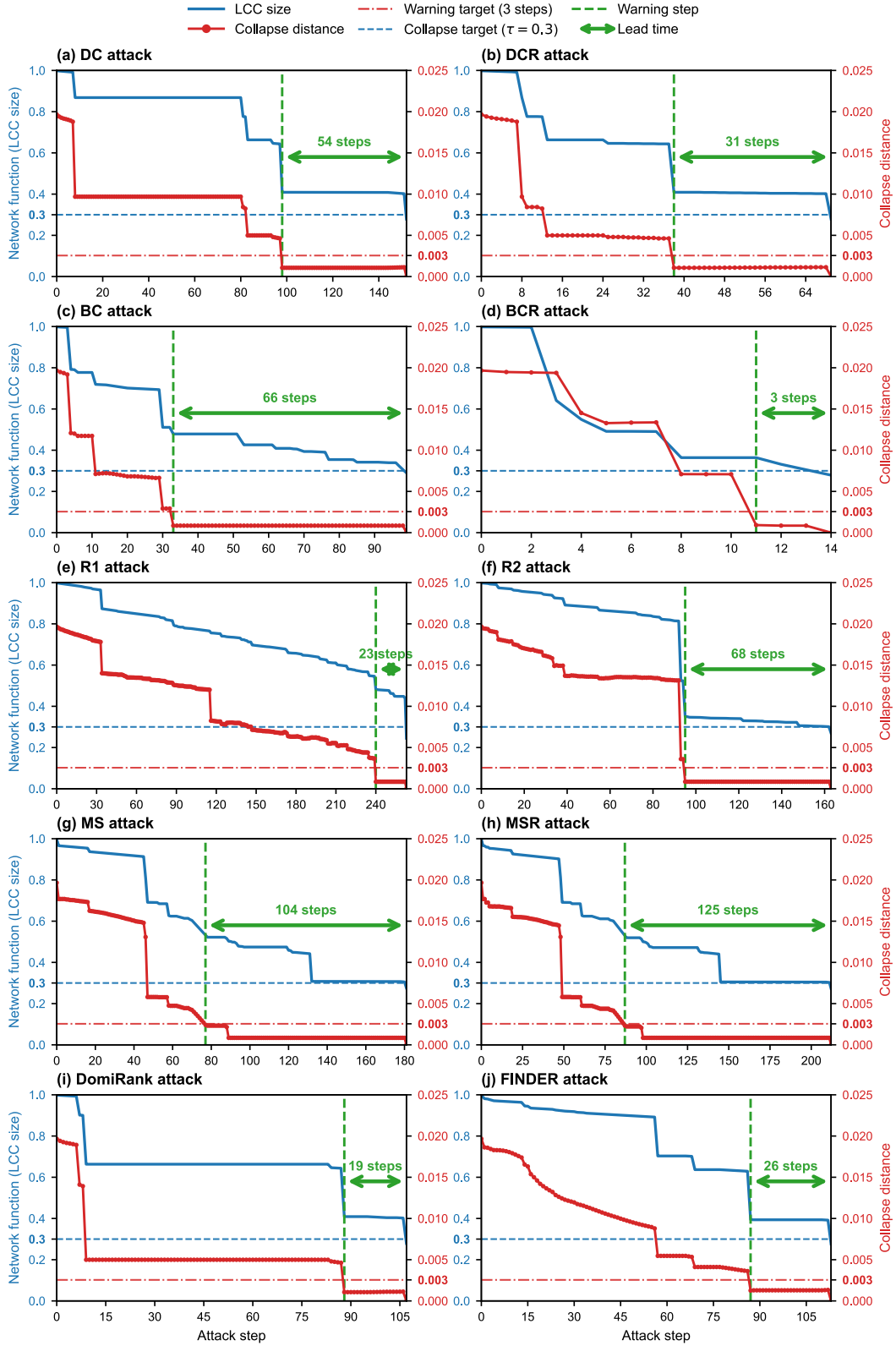


Figure S13. Early-warning signal of the Power network under 30 attack scenarios at $\tau = 0.3$.

Plot elements are the same as in Fig. 6.

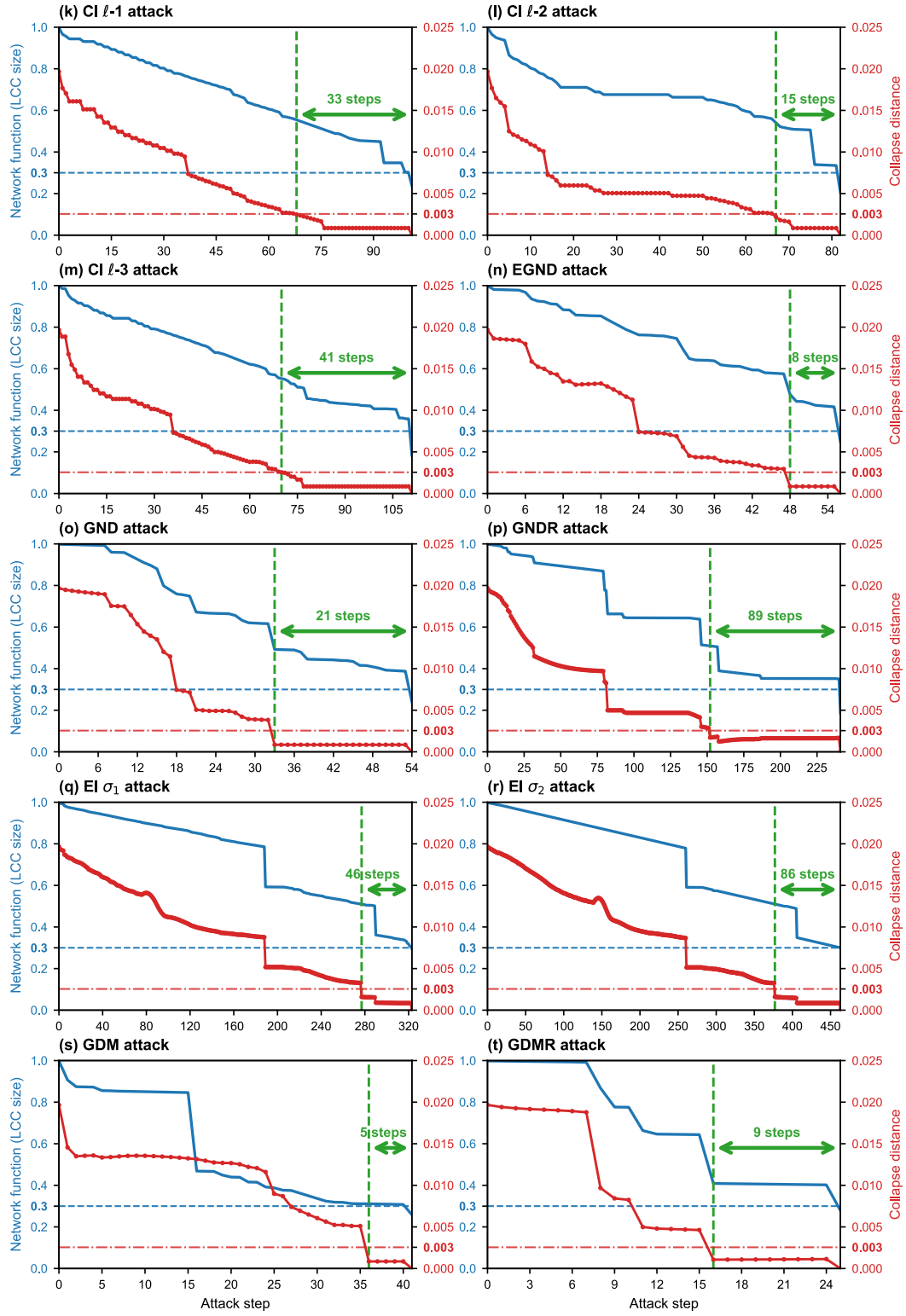


Figure S13 (continued).

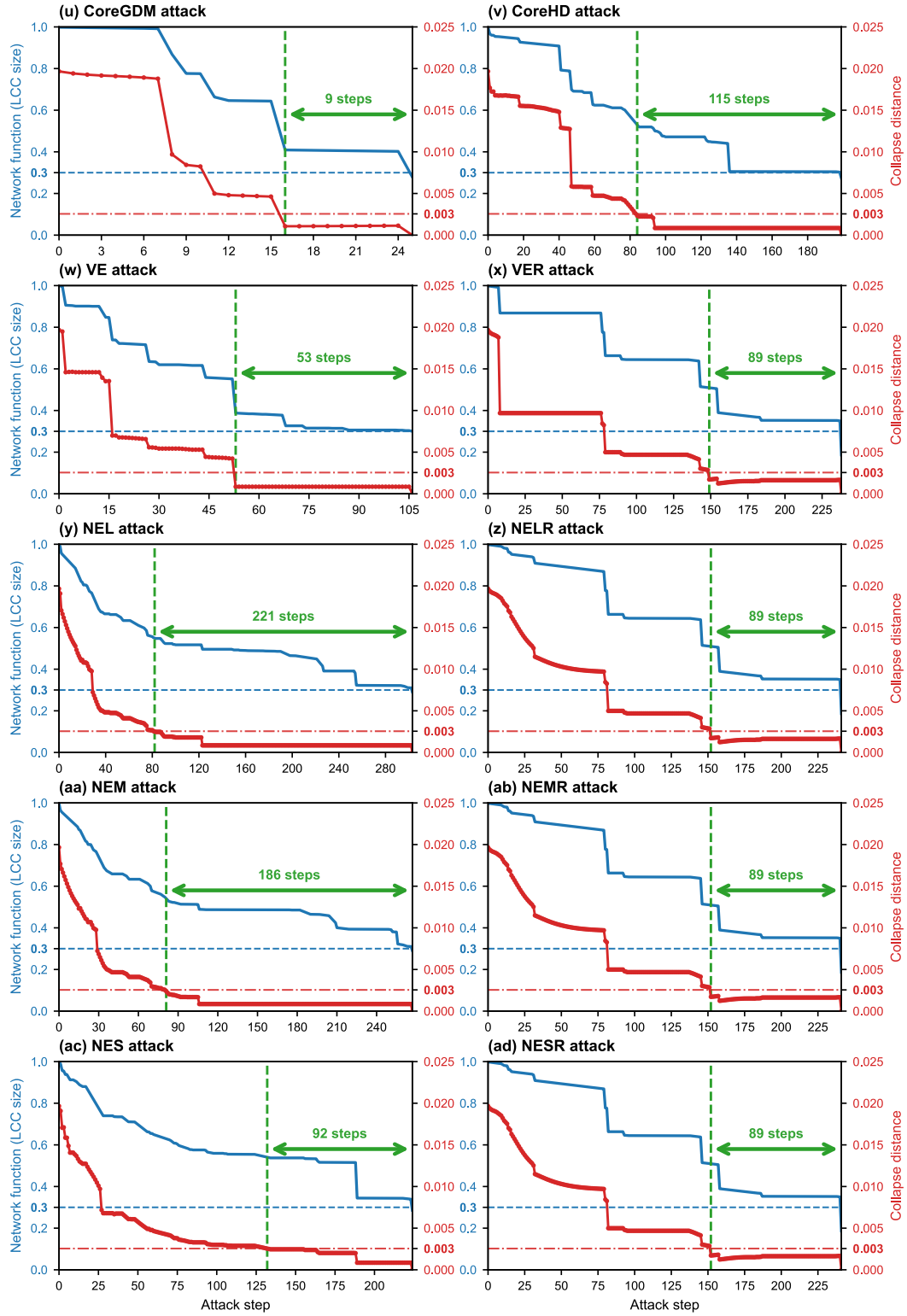


Figure S13 (continued).

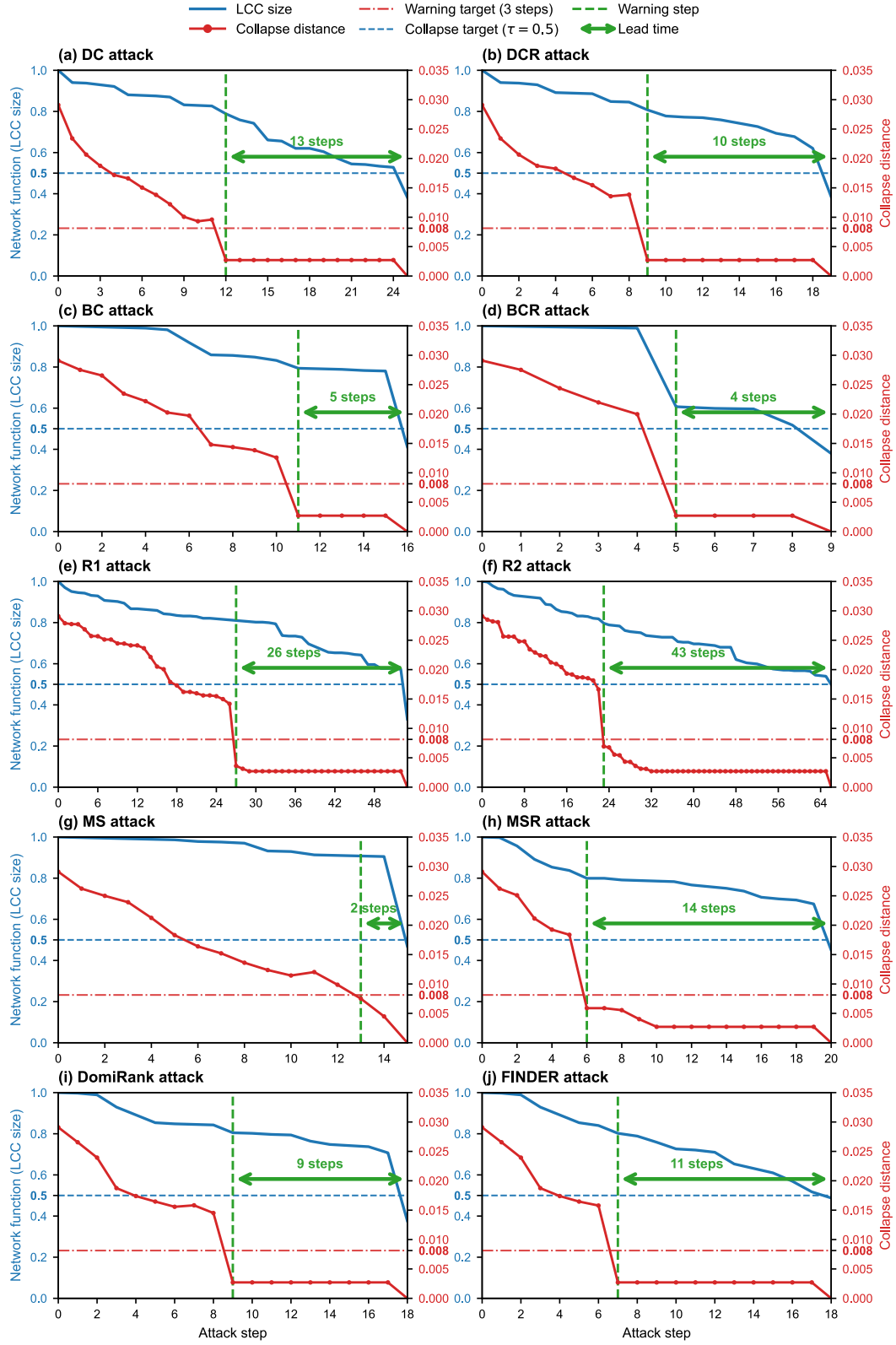


Figure S14. **Early-warning signal of the Transport network under 30 attack scenarios at $\tau = 0.5$.**

Plot elements are the same as in Fig. 6.

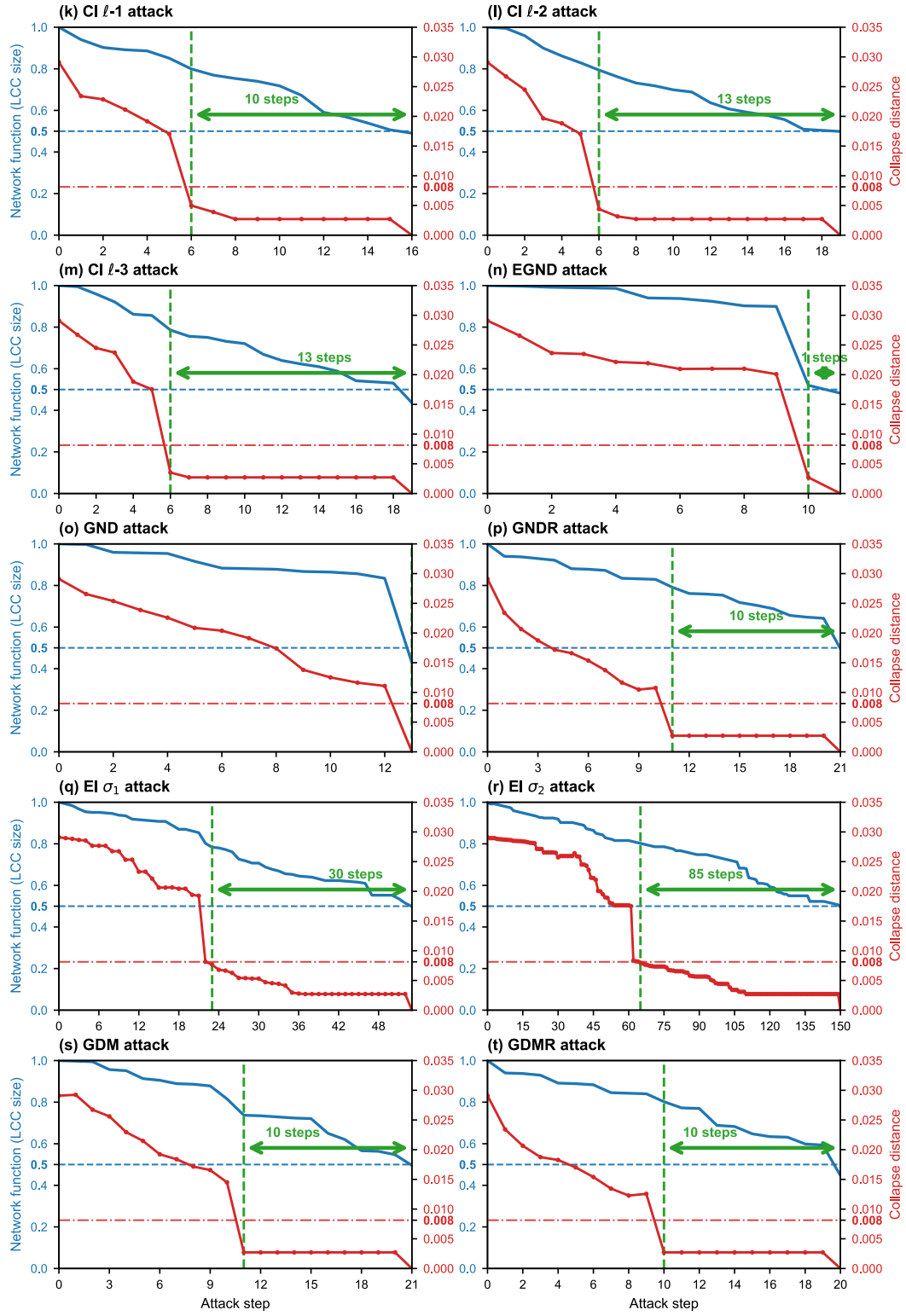


Figure S14 (continued).

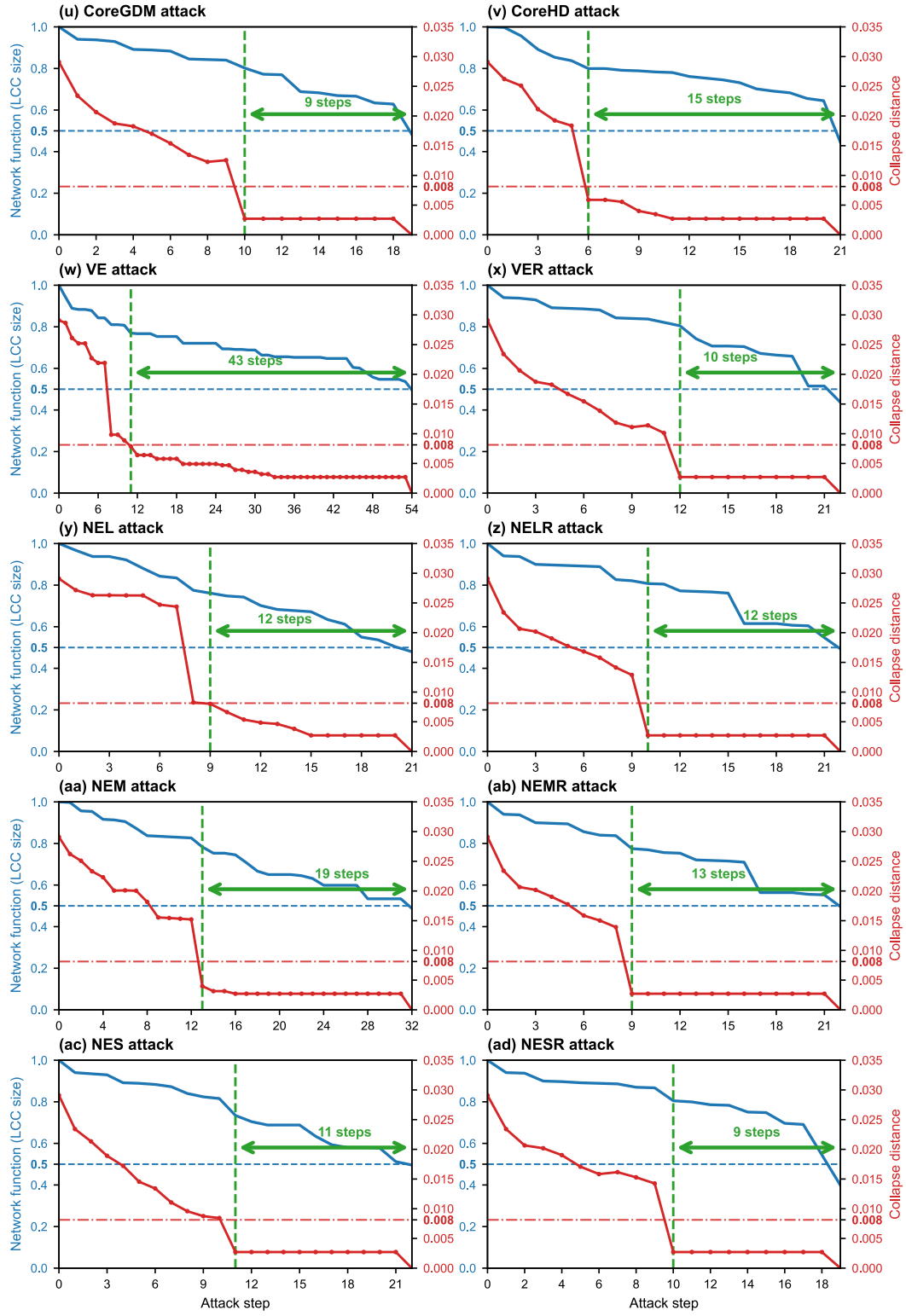


Figure S14 (continued).

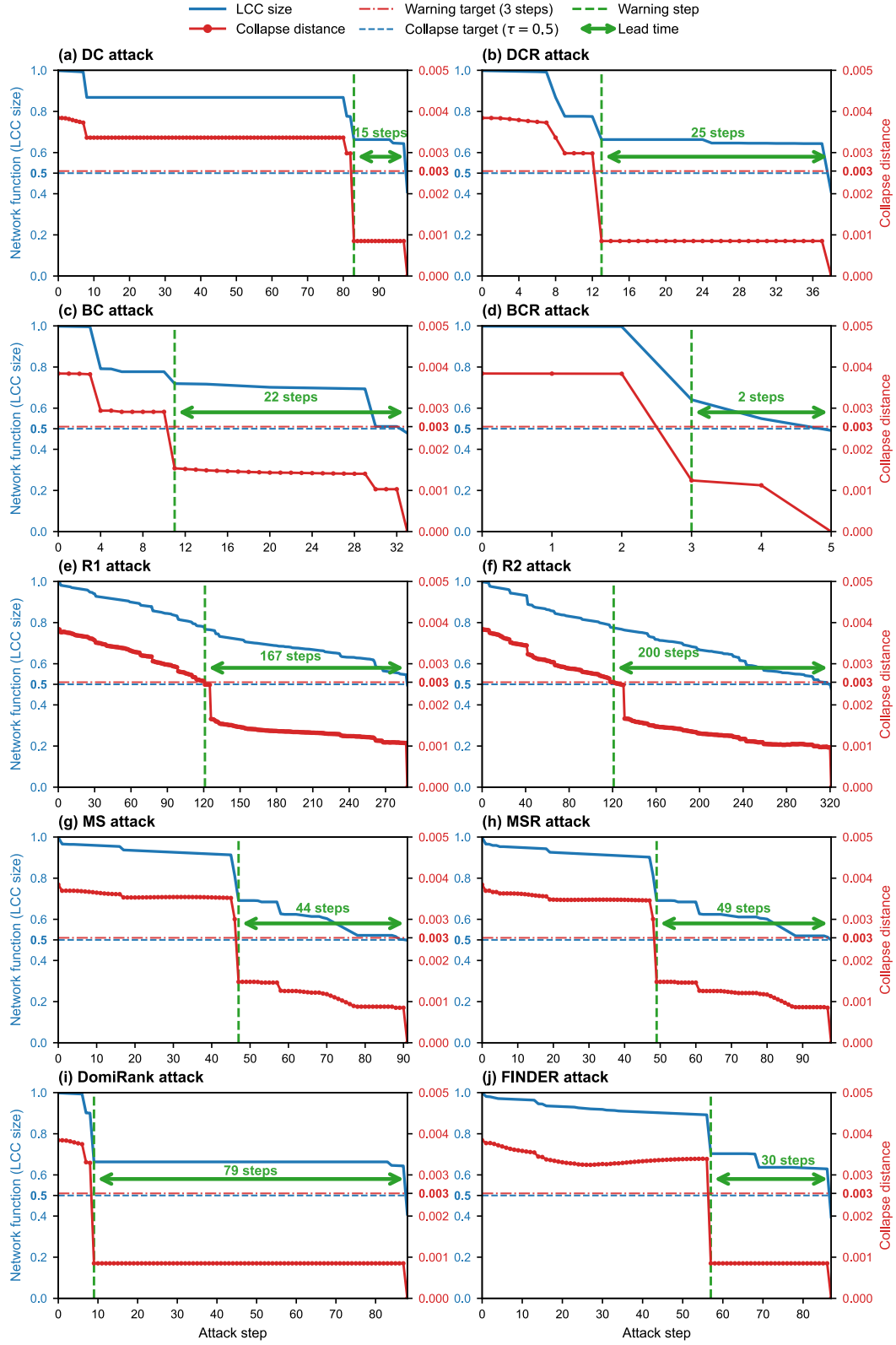


Figure S15. Early-warning signal of the Power network under 30 attack scenarios at $\tau = 0.5$.

Plot elements are the same as in Fig. 6.

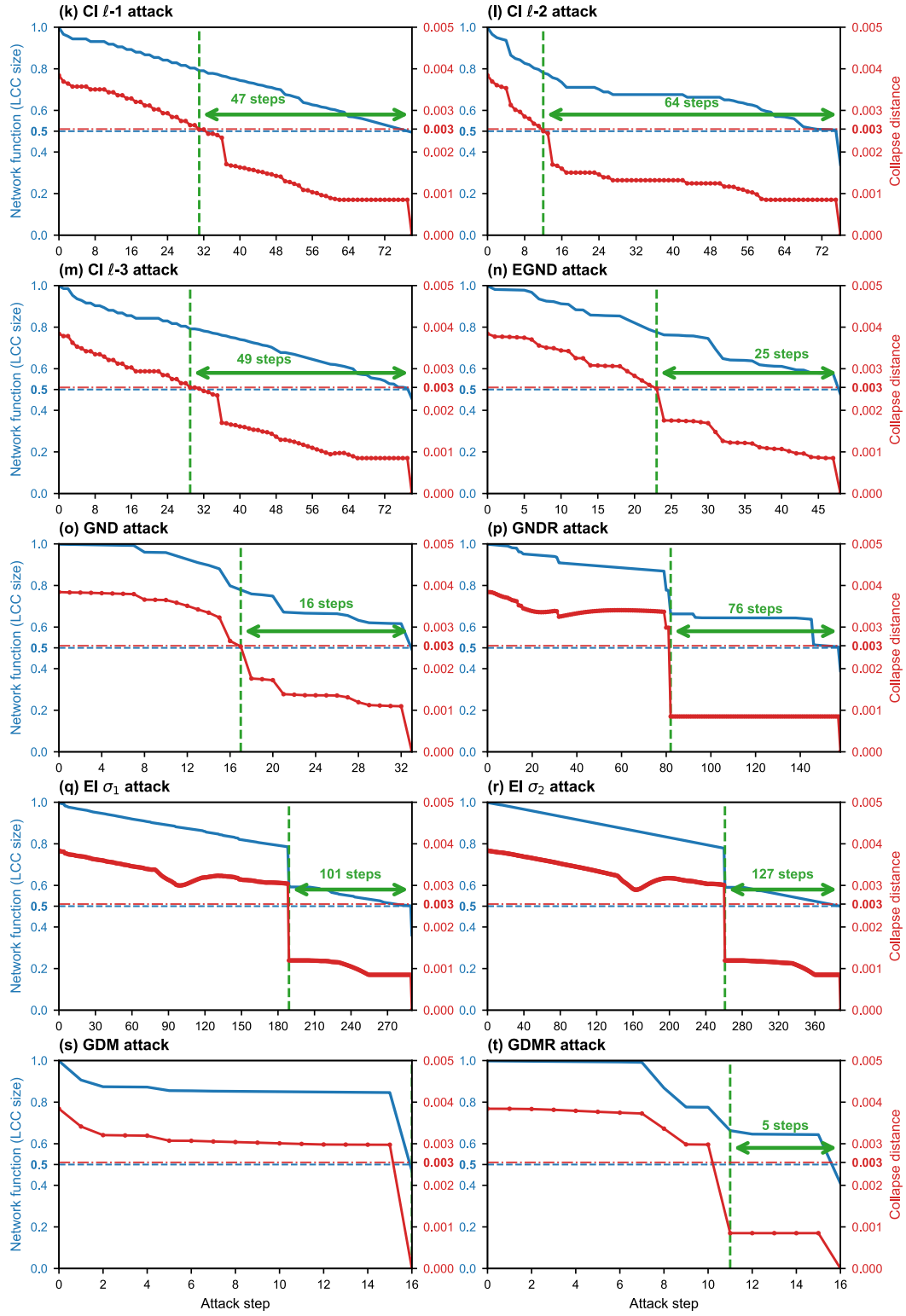


Figure S15 (continued).

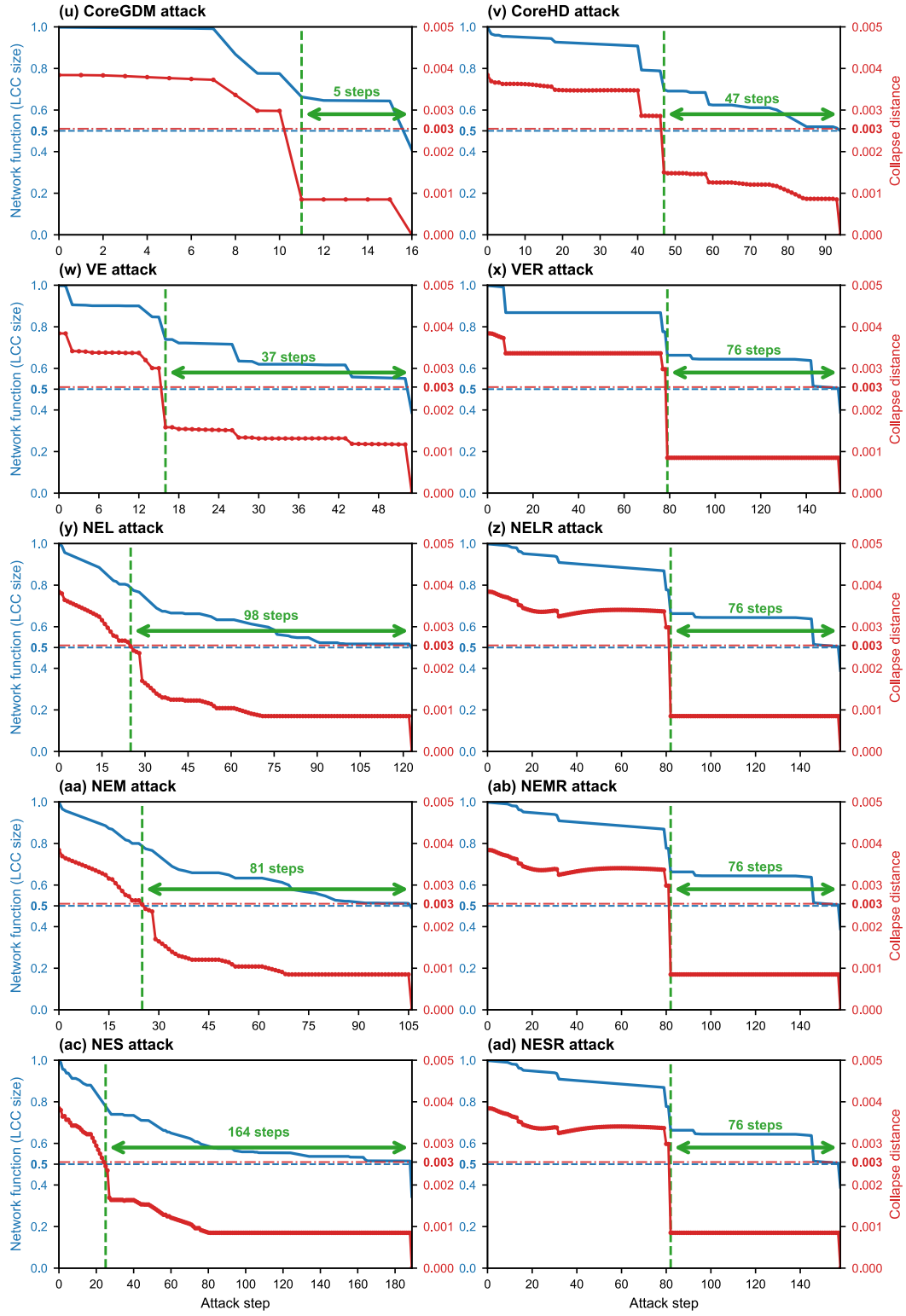


Figure S15 (continued).

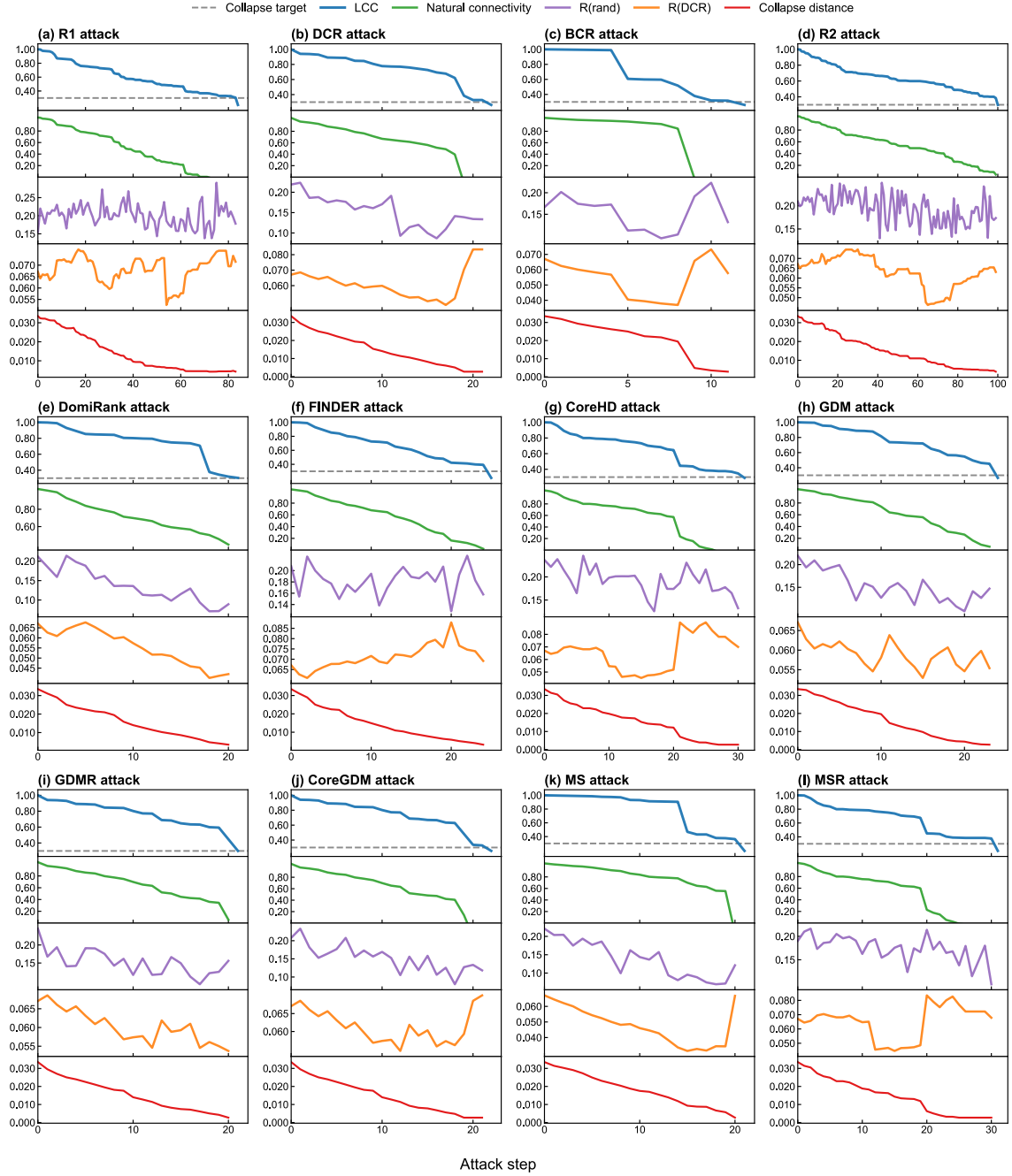


Figure S16. **Online multi-metric trajectories of the Transport network under sustained attack at $\tau = 0.3$.** From top to bottom in each panel, we show LCC (dashed horizontal line indicates τ), natural connectivity $\bar{\lambda}$, $R(\text{rand})$, $R(\text{DCR})$, and the collapse distance estimated by TCR-GIN, as functions of attack step. The LCC trajectory is shown through the collapse step, whereas the other metrics are plotted only up to the last pre-collapse snapshot.

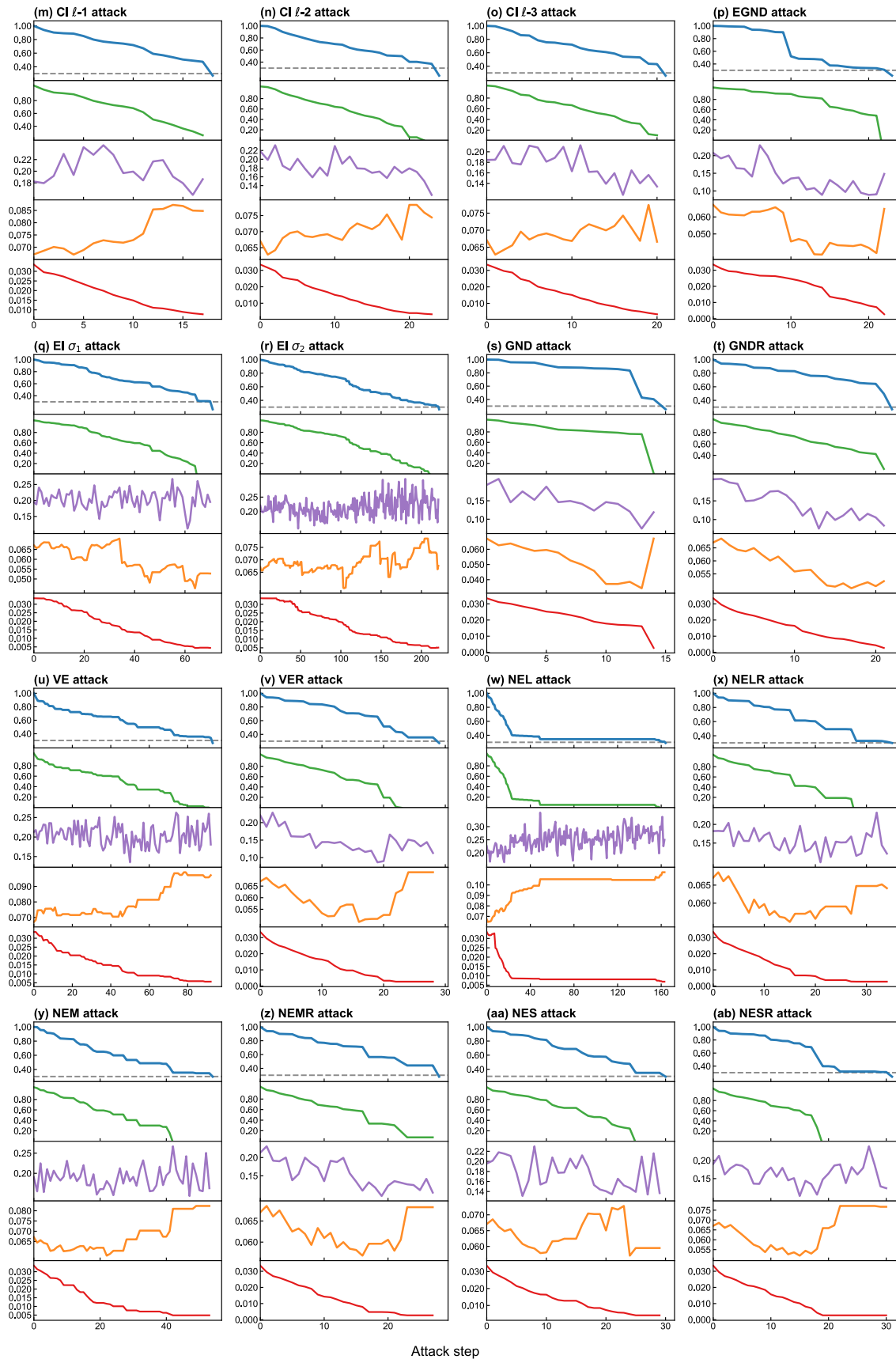


Figure S16 (continued).

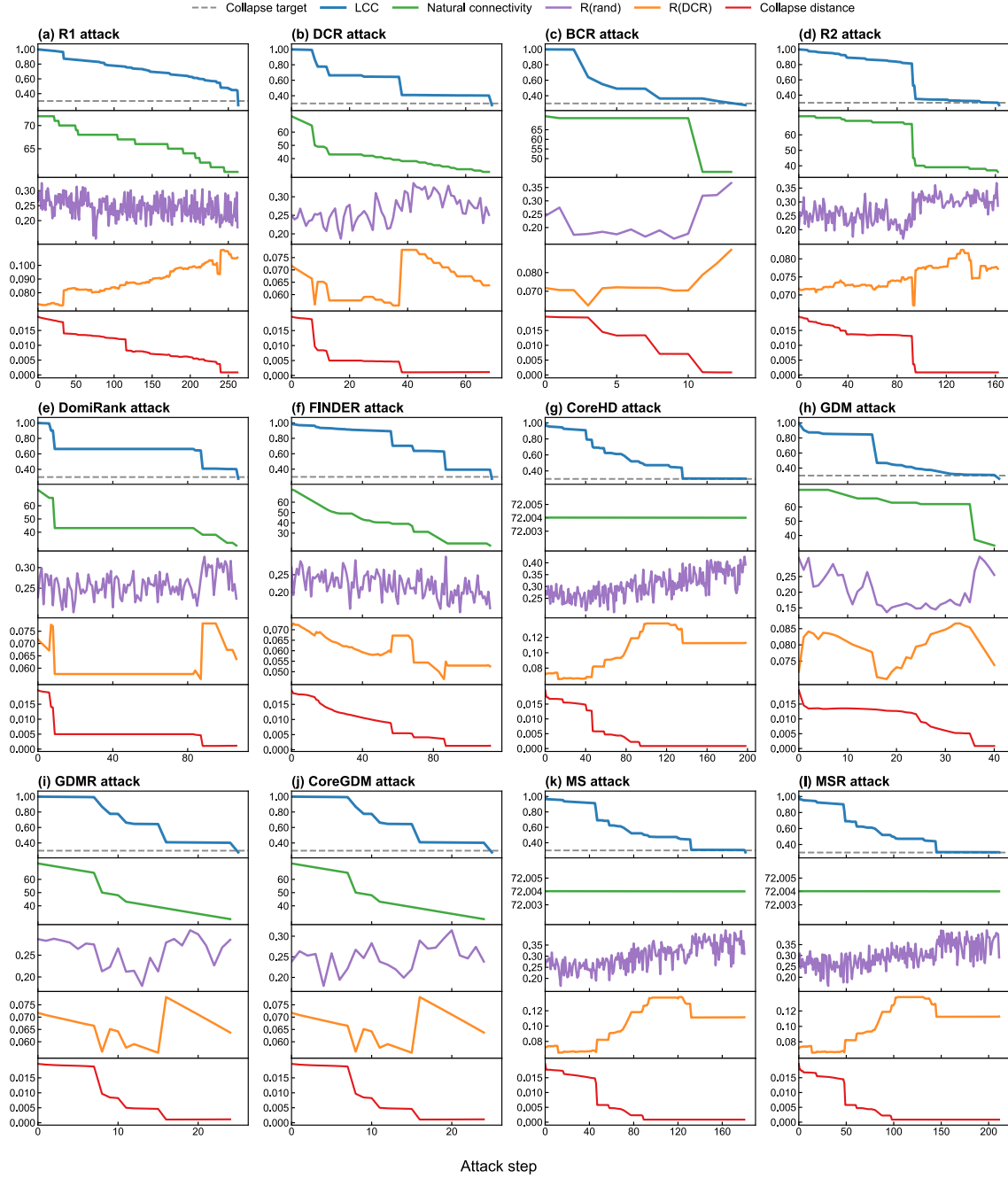


Figure S17. **Online multi-metric trajectories of the Power network under sustained attack at $\tau = 0.3$.** From top to bottom in each panel, we show LCC (dashed horizontal line indicates τ), natural connectivity $\bar{\lambda}$, $R(\text{rand})$, $R(\text{DCR})$, and the collapse distance estimated by TCR-GIN, as functions of attack step. The LCC trajectory is shown through the collapse step, whereas the other metrics are plotted only up to the last pre-collapse snapshot.

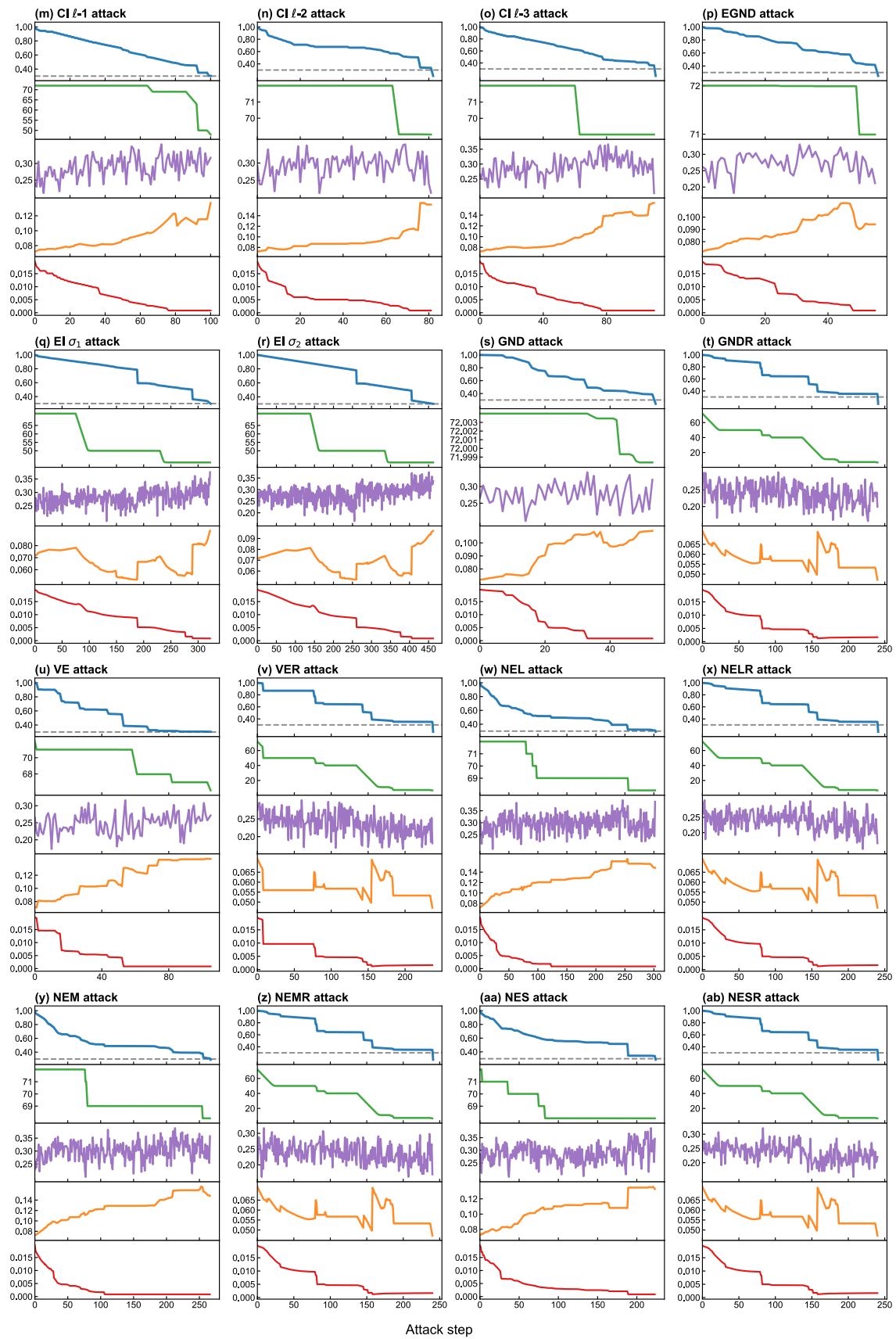


Figure S17 (continued).

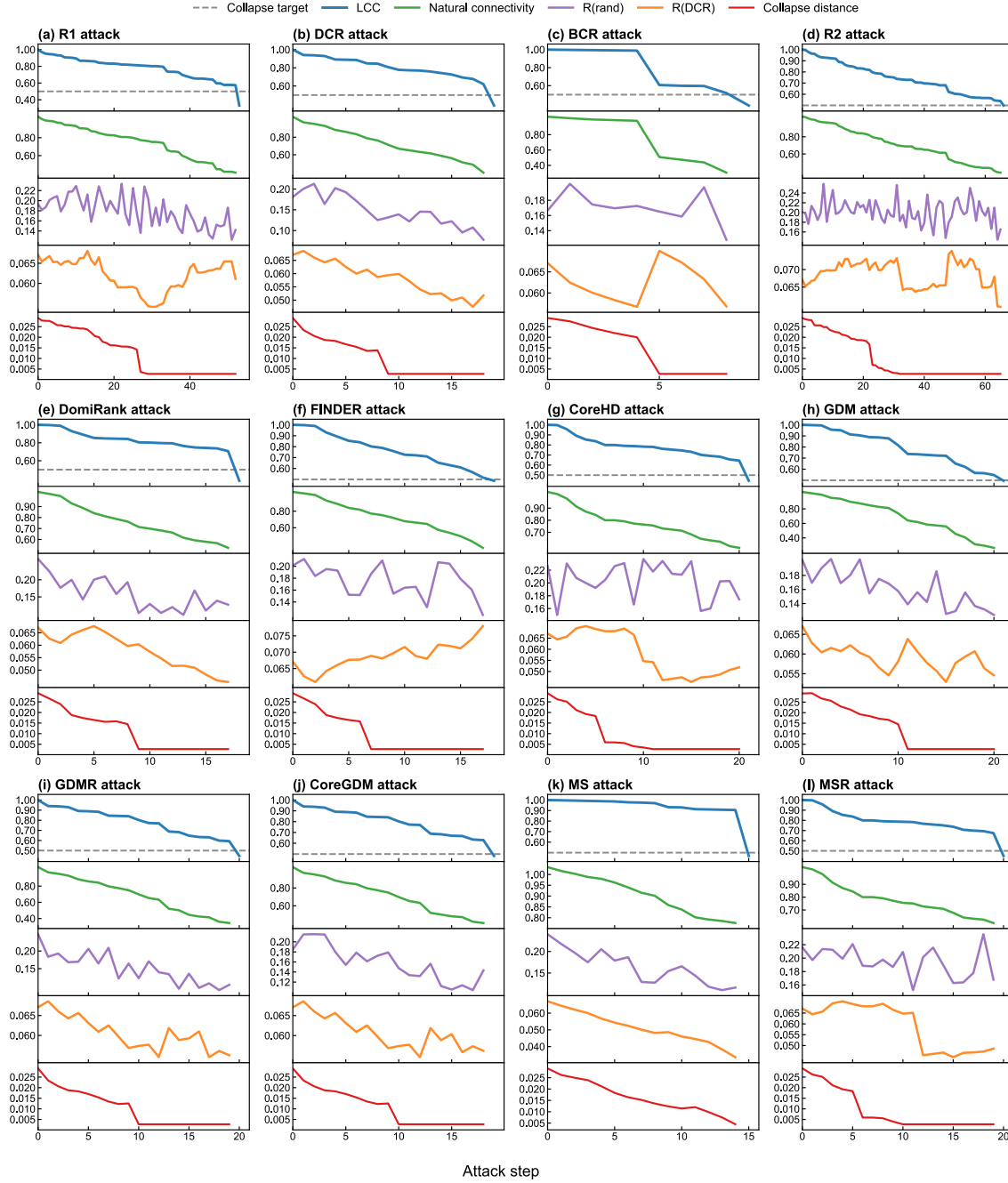


Figure S18. **Online multi-metric trajectories of the Transport network under sustained attack at $\tau = 0.5$.** From top to bottom in each panel, we show LCC (dashed horizontal line indicates τ), natural connectivity $\bar{\lambda}$, $R(\text{rand})$, $R(\text{DCR})$, and the collapse distance estimated by TCR-GIN, as functions of attack step. The LCC trajectory is shown through the collapse step, whereas the other metrics are plotted only up to the last pre-collapse snapshot.

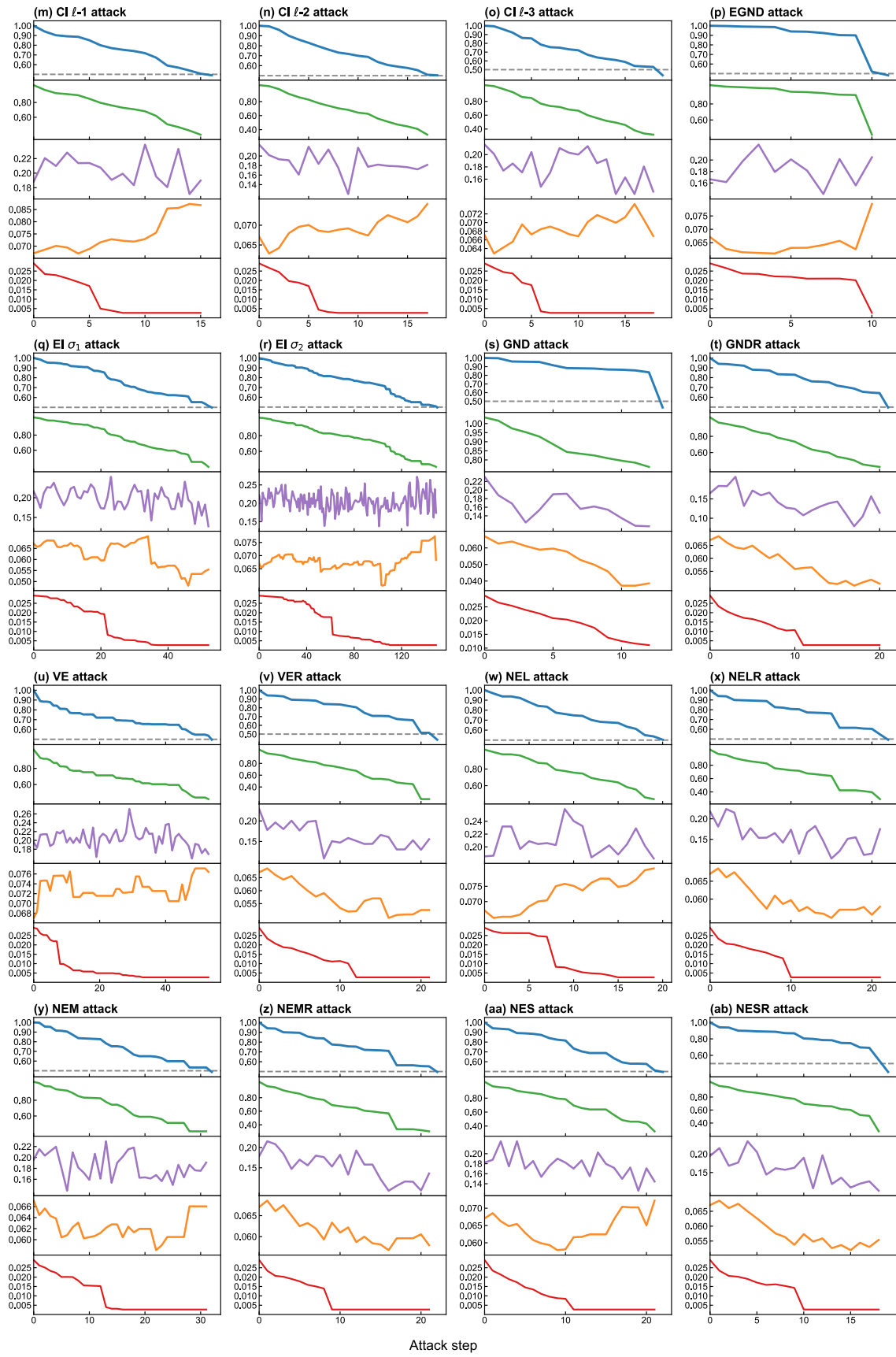


Figure S18 (continued).

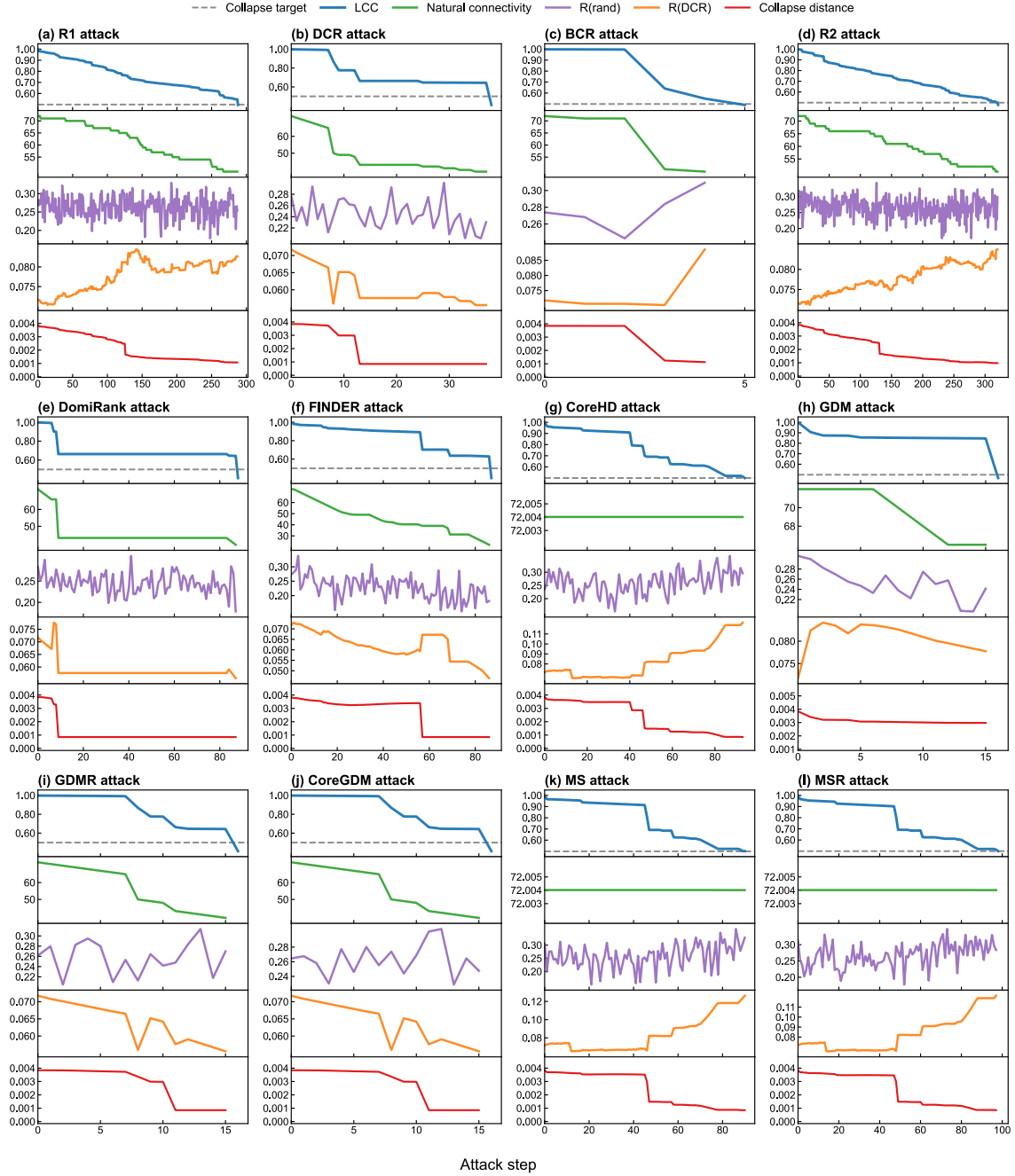


Figure S19. **Online multi-metric trajectories of the Power network under sustained attack at $\tau = 0.5$.** From top to bottom in each panel, we show LCC (dashed horizontal line indicates τ), natural connectivity $\bar{\lambda}$, $R(\text{rand})$, $R(\text{DCR})$, and the collapse distance estimated by TCR-GIN, as functions of attack step. The LCC trajectory is shown through the collapse step, whereas the other metrics are plotted only up to the last pre-collapse snapshot.

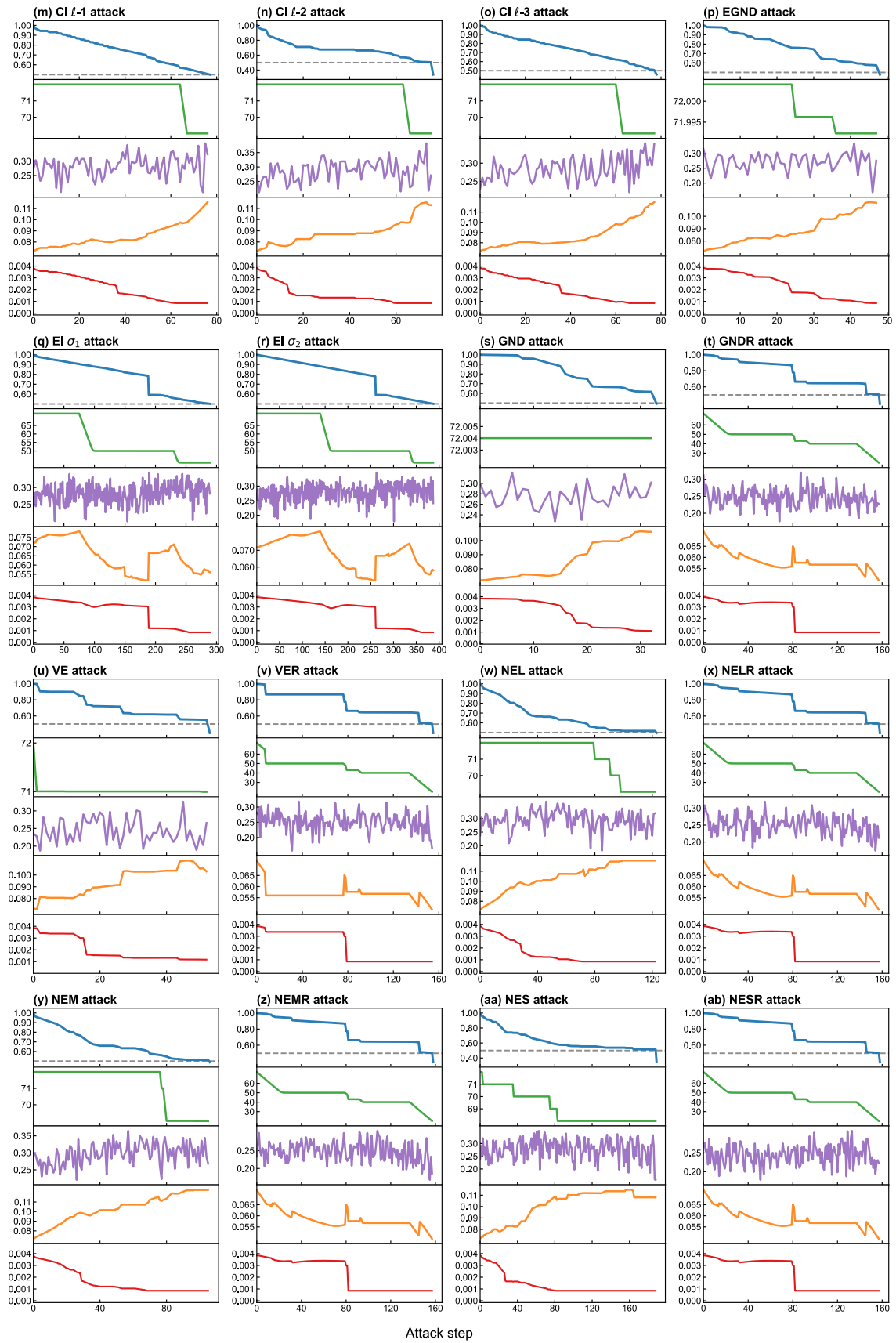


Figure S19 (continued).

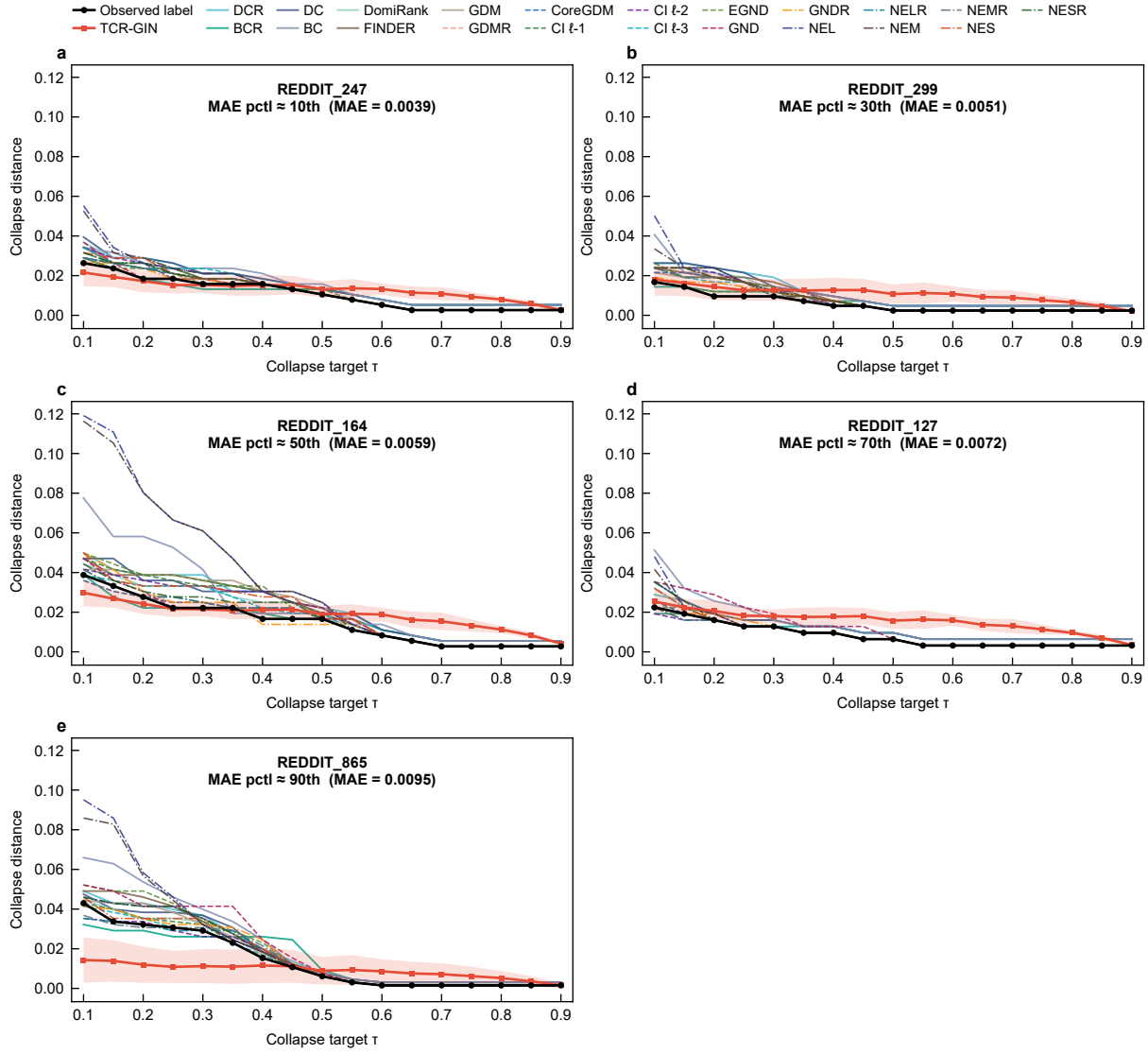


Figure S20. **Representative collapse-profile predictions on REDDIT test networks.** Collapse distance is evaluated over a discrete collapse-target grid $\tau = 0.1, \dots, 0.9$. Panels (a–e) show five representative test networks selected according to the mean absolute error (MAE) percentile of the TCR-GIN profile relative to the observed-label profile, corresponding approximately to the 10th, 30th, 50th, 70th, and 90th percentiles, respectively. The black curve denotes the observed-label profile, while the red curve denotes the TCR-GIN prediction. Other colored curves show the collapse-distance profiles obtained from individual heuristic baselines. Across a broad range of profile errors, TCR-GIN captures the overall target-dependent trend of collapse distance, illustrating that the scalar CD estimator can be extended to a lightweight vector-valued collapse-profile predictor.

-
- [1] Callaway, D. S., Newman, M. E., Strogatz, S. H. & Watts, D. J. Network robustness and fragility: Percolation on random graphs. *Physical review letters* **85**, 5468 (2000).
- [2] Morone, F. & Makse, H. A. Influence maximization in complex networks through optimal percolation. *CoRR* **abs/1506.08326** (2015). 1506.08326.
- [3] Jun, W., Barahona, M., Yue-Jin, T. & Hong-Zhong, D. Natural connectivity of complex networks. *Chinese Physics Letters* **27**, 078902 (2010).
- [4] Schneider, C. M., Moreira, A. A., Andrade Jr, J. S., Havlin, S. & Herrmann, H. J. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences* **108**, 3838–3841 (2011).
- [5] Artime, O. *et al.* Robustness and resilience of complex networks. *Nature Reviews Physics* **6**, 114–131 (2024).
- [6] Engsig, M., Tejedor, A., Moreno, Y., Foufoula-Georgiou, E. & Kasmi, C. Domirank centrality reveals structural fragility of complex networks via node dominance. *Nature Communications* **15** (2024).
- [7] Freeman, L. C. Centrality in social networks conceptual clarification. *Social networks* **1**, 215–239 (1978).
- [8] Morone, F., Min, B., Bo, L., Mari, R. & Makse, H. A. Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific Reports* **6** (2016).
- [9] Ghavasieh, A., Stella, M., Biamonte, J. D. & Domenico, M. D. Unraveling the effects of multiscale network entanglement on disintegration of empirical systems. *CoRR* **abs/2008.05368** (2020). 2008.05368.
- [10] Huang, Y., Wang, H., Ren, X.-L. & Lü, L. Identifying key players in complex networks via network entanglement. *Communications Physics* **7** (2024).
- [11] Ren, X.-L., Gleinig, N., Helbing, D. & Antulov-Fantulin, N. Generalized network dismantling. *Proceedings of the national academy of sciences* **116**, 6554–6559 (2019).
- [12] Ren, X.-L. & Antulov-Fantulin, N. Ensemble approach for generalized network dismantling. In *International Conference on Complex Networks and Their Applications*, 783–793 (Springer, 2019).
- [13] Braunstein, A., Dall’Asta, L., Semerjian, G. & Zdeborová, L. Network dismantling. *CoRR* **abs/1603.08883** (2016). 1603.08883.
- [14] Zdeborová, L., Zhang, P. & Zhou, H.-J. Fast and simple decycling and dismantling of networks. *Scientific Reports* **6** (2016).

- [15] Clusella, P., Grassberger, P., Pérez-Reche, F. J. & Politi, A. Immunization and targeted destruction of networks using explosive percolation. *Physical Review Letters* **117** (2016).
- [16] Fan, C., Zeng, L., Sun, Y. & Liu, Y.-Y. Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence* **2**, 317–324 (2020).
- [17] Grassia, M., De Domenico, M. & Mangioni, G. Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nature Communications* **12** (2021).
- [18] Grassia, M. & Mangioni, G. Coregdm: Geometric deep learning network decycling and dismantling. In *International Workshop on Complex Networks*, 86–94 (Springer, 2023).
- [19] Wu, C. *et al.* A multitask network robustness analysis system based on the graph isomorphism network. *IEEE Trans. Cybern.* **54**, 6630–6642 (2024).
- [20] De Domenico, M., Solé-Ribalta, A., Gómez, S. & Arenas, A. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences* **111**, 8351–8356 (2014).
- [21] Rossi, R. & Ahmed, N. The network data repository with interactive graph analytics and visualization. *Proceedings of the AAAI Conference on Artificial Intelligence* **29** (2015).
- [22] Leskovec, J., Kleinberg, J. & Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)* **1**, 2-es (2007).
- [23] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
- [24] Hu, W. *et al.* Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* **33**, 22118–22133 (2020).