Supplemental Material for "High-order Knowledge Based Network Controllability Robustness Prediction: A Hypergraph Neural Network Approach"

Shibing Mo, Jiarui Zhang†, Jiayu Xie†, Xiangyi Teng, *Member, IEEE*, and Jing Liu, *Senior Member, IEEE*.

I. Supplementary Materials

A. Related Work

1) The application of hypergraph neural networks: Hypergraph neural networks — a term used in the literature with several related meanings (e.g., supernetworks that embed many candidate architectures, graph-of-graphs or hypergraph representations where each node is itself a graph, and extensions of hypergraph/hypergraph laplacians) — have already been applied across a remarkably broad set of domains and tasks. In neural architecture search, the differentiable search paradigm treats the search space as a single over-arching hypergraph (or supernetwork) that contains all candidate cells/paths and is optimized end-to-end, enabling efficient architecture optimization for image classification and related tasks [1]. In problems that compare or index whole graphs, researchers build a hypergraph of graphs (each node represents a data graph and edges encode pairwise similarities) so that graph-level classification, retrieval, or similarity search can be cast as node classification/lookup on the hypergraph; this idea underpins recent graph-graph similarity networks and neural hypergraph containment/search systems designed for large graph databases and efficient hypergraph queries [2], [3], [3]. In neuroscience and medical imaging, hypergraph and graph-super-resolution methods use graph neural models to upsample or refine low-resolution brain connectivity maps into higher-resolution connectomes, improving downstream analyses that require fine-grained regional networks [4]. Engineering applications have also adopted hypergraph-style architectures: spatial–temporal hypergraph feature extractors and multi-scale hypergraph convolutional modules have been proposed for rotating machinery fault diagnosis and structural health monitoring, where mapping multi-sensor signals into graphs and then into hypergraph representations yields robust features for anomaly detection [5], [6]. Additionally, the interactive concept of supergraph neural networks can be applied to integrate multi-modal and multi-feature scenarios in single-cell multi-omics [7], [8]. In image processing, graph-based and hypergraph-inspired GNNs support multi-image and cross-scale super-resolution / restoration tasks by modeling nonlocal self-similarities across patches as graph nodes and using graph message passing to aggregate complementary information from multiple observations [9]. Recommender systems and social recommendation have benefited from hypergraph and hypergraph constructions (often implemented via HGNN [5] variants) that explicitly model multi-party interactions, social groups, and multi-view modalities to better capture group influence and high-order affinities; empirical studies report improved recommendation and robustness when hypergraph/hypergraph attention mechanisms are used [10], [11]. From a systems and scalability perspective, the hypergraph abstraction is useful for distributed or federated GNN training: constructing a hypergraph to summarize or index many local graphs (or to cluster nodes/partitions) can reduce communication and speed up training on massive graph collections [12]. Beyond these concrete applications, surveys and reviews of hypergraph and hypergraph methodologies highlight how hypergraph laplacians, hypergraph generalizations, and hypergraph kernels provide principled theoretical foundations that have catalyzed diverse applications in multimodal fusion, biological network analysis, document and text classification, and knowledge-graph style queries; these reviews also point to ongoing research on efficiency, attention-based weighting of hyperedges, and multiset / AllSet [13] style propagation rules that extend the expressive power of conventional GNNs [14], [15]. Finally, recent work on neural similarity search, graph coding for efficient hypergraph search, and specialized index structures shows that hypergraph neural methods are not only beneficial for predictive modeling but also for practical graph-database operations such as containment search and large-scale retrieval, making them attractive for industrial graph-mining, cheminformatics, and program analysis pipelines [3], [16], [17].

B. Formalizing high-order knowledge

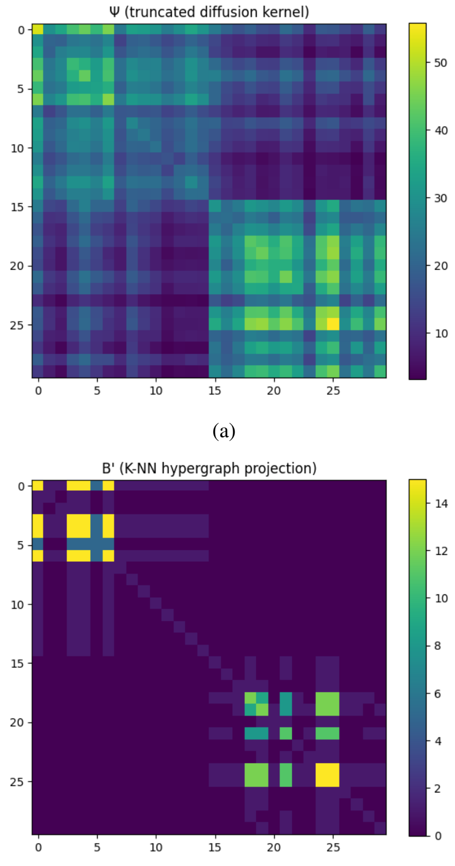
This subsection gives a concise formal definition of high-order knowledge, states two propositions that make precise why hypergraph-derived features are strictly richer than 1-hop (pairwise) features for many controllability/robustness targets, and explains how these results connect to the Dual HGNN formulas **??**-**??**.

1) Notation: Let $G = (V, E)$ be a simple graph with $|V| = n$ nodes and adjacency matrix $A \in \{0,1\}^{n\times m}$. For an integer $K \geq 1$ denote the set of walk matrices $\{A, A^2, \ldots, A^K\}$. For a hypergraph constructed on V with m hyperedges, let $H \in \{0,1\}^{n\times m}$ be its node-hyperedge incidence matrix and let $W \in \mathbb{R}^{m\times m}$ be a diagonal matrix of hyperedge weights. Define the node-projection matrix $B := HWH^T \in \mathbb{R}^{n\times n}$. When needed, we write entrywise relations as B_{ij}. We denote by $\mathcal{A}_K$ the algebra generated by the matrices $\{I, A, A^2, \ldots, A^K\}$ under linear combinations and (optional) elementwise boolean thresholding.

2) High-order knowledge Definition: A feature object (matrix/tensor/collection) $\mathcal{H}$ computed from G is said to encode **high-order knowledge of order** K if $\mathcal{H}$ depends nontrivially on walk information of length up to K; equivalently,

- $\mathcal{H}$ lies in the algebra generated by $\{A, A^2, \ldots, A^K\}$ but not in the subalgebra generated by $\{I, A\}$ alone, or
- $\mathcal{H}$ is produced by a hypergraph incidence H whose construction uses K-ball / K-step relations (e.g., K-Hop hyperedges) or a kernel/diffusion $\Psi(A)$ incorporating powers up to degree K (e.g., embeddings derived from spectral/propagation maps).

Intuitively, high-order refers to information that cannot be recovered from only immediate-neighbor (1-hop) relations.



(a)

(b)

Fig. S1. Visualization results of the corresponding matrix to compare large entries of ψ and B' (with top pairs overlapping).

3) Propositions: We state two propositions that formalize the intuition that K-Hop and K-NN hypergraphs (as used in this work) encode strictly more information than 1-hop adjacency for many targets of interest.

i) *Proposition 1 (K-Hop hypergraph sufficiency).* Let $HG_{K\text{-Hop}}$ be the hypergraph constructed by taking, for each node $v \in V$, the K-ball $\mathcal{N}_K(v) := \{u : \exists k \leq K, (A^k)_{uv} > 0\}$ as a hyperedge (or by using a collection of such balls), and let H be its incidence matrix. Then the node-projection matrix $B = HWH^T$ is a function of the family $\{A, A^2, \ldots, A^K\}$. Consequently, under (Assum1) there exists a map $\mathcal{F}$ (possibly nonlinear) such that $R(G) = g(\sum_{k=1}^{K} \alpha_k A^k) = g(\mathcal{F}(B))$. Hence any predictor that has access to B (or learnable transforms of B) can in principle recover the dependence of $R(G')$ expressed in (Assum1).

Proof (sketch). For each center v define the indicator vector $\mathbf{1}_{\mathcal{N}_K(v)} \in \{0,1\}^n$ where the i-th entry equals 1 iff there exists a walk of length $\leq K$ from v to i; equivalently this indicator depends only on the booleanized sum $\mathbf{1}(\sum_{k=1}^{K} A^k > 0)$. If the hyperedge set contains the K-balls (one per center) and weights in W are chosen deterministically (e.g., all ones), then $B = \sum_{v \in V} \mathbf{1}_{\mathcal{N}_K(v)} \mathbf{1}_{\mathcal{N}_K(v)}^T$, which shows that each entry B_{ij} is a count of how many K-balls contain both i and j. Therefore every entry of B is a function of the booleanized matrices $\{(\mathbf{1}(A^k > 0))\}_{k \leq K}$ (and hence of $\{A^k\}_{k \leq K}$ up to thresholding). As the right-hand side in (Assum1) depends only on $\{A^k\}_{k \leq K}$, there exists a (possibly many-to-one) map $\mathcal{F}$ from B to $\sum_{k \leq K} \alpha_k A^k$ (or to sufficient statistics for g). Thus the stated equality follows.

Remark. Exact algebraic invertibility (i.e., uniquely recovering A^k from B) is not required: it suffices that the information in B contains the sufficient statistics used by g. In practice, the Dual HGNN learns $\mathcal{F}$ or an approximation thereof.

ii) *Proposition 2 (K-NN hypergraph and diffusion kernels).* Let $\bar{\Phi} : V \to \mathbb{R}^d$ be node embeddings such that the pairwise similarity matrix S with entries $S_{ij} = \langle \bar{\Phi}(i), \bar{\Phi}(j) \rangle$ approximates a diffusion kernel of the adjacency, $S \approx \Psi(A) := \sum_{k=0}^{\infty} \beta_k A^k$, with the truncation $\sum_{k=0}^{K} \beta_k A^k$ dominating contributions up to order K. Constructing a K-NN hypergraph by linking each node to its K_{NN} nearest neighbors in embedding space yields a projection matrix B' whose large entries correspond to large entries of $\Psi(A)$. If $R(G)$ depends on structural equivalence or diffusion-based proximity (i.e. on $\Psi(A)$), then B' is informative for predicting $R(G)$ while 1-hop adjacency alone is not.

Proof (sketch).

Under the approximation $S \approx \Psi(A)$ assume the approximation error is small in operator norm on the subspace of interest. Thresholding or taking the K-NN of S selects pairs with the largest values of $\Psi(A)_{ij}$, which reflect sums of weighted walk counts between nodes up to order K. The resulting incidence matrix H' and projection $B' = H'W'H'^T$ therefore encode (thresholded) diffusion information and role-similarity that go beyond immediate adjacency. If $R(G)$ depends on such diffusion or role-based features, a model using B' can access them while a model restricted to functions of A with degree ≤ 1 cannot, concluding the claim.

Besides, we provide some numerical analysis as follows:

Fig. S1 (a) ψ represents the node embedding ψ (spectral embedding) generated by extracting the eigenvector of the diffusion kernel with a truncation order of 40. The Fig. S1 (b) visualizes the projection matrix B' computed after constructing a hypergraph using K-NN.

The heatmap of ψ displays larger diffusion affinity entries within two communities (two blocks of the matrix). These high values result from the accumulation of paths spanning multiple steps. The heatmap of B' reveals the "co-occurrence" structure preserved after K-NN hypergraph projection: most node pairs frequently co-occurring within the same hyperedge manifest as large values in B'. The top-40 overlap rate ($\approx 45\%$) indicates

TABLE S1
BC_{GAT} MODEL NETWORK STRUCTURE

Layer Name	Layer Type	Input Channels	Output Channels	Number of Parameters
gat1	GATConv (4 heads)	1	64	$1 \times 64 \times 4 + 2 \times 64 \times 4 + 64 \times 4 =$ **1024**
gat2	GATConv (1 head)	$64 \times 4 = 256$	64	$256 \times 64 + 2 \times 64 + 64 =$ **16576**
fc	Fully Connected	64	1	$64 \times 1 + 1 =$ **65**

that under this setting, the K-NN hypergraph indeed captures a portion of ψ's major entries (particularly node pairs that are significantly close in the embedding space).

4) Connection to Dual HGNN: The Dual HGNN processes hypergraph structure via two alternating steps: node-to-hyperedge aggregation and hyperedge-to-node aggregation (formulas **??**-**??**). These steps are learnable functions of the incidence matrix H and node/hyperedge features. Concretely, the message passing performed by the Dual HGNN can be written schematically as

$$X_E \leftarrow f_E(H^T, X), \quad X \leftarrow f_V(H, X_E),$$

where X and X_E are node and hyperedge feature matrices and each f includes attention weights that are learned (see formulas **??**-**??**). Because the node-projection matrix $B = HWH^T$ is a sufficient statistic for the high-order relations described above (Propositions 1–2), the Dual HGNN has the representational capacity to learn maps of the form $X_{\text{out}} = \text{NN}(\mathcal{T}(B, X)) \approx \text{NN}(\mathcal{T}'(\{A^k\}_{k \leq K}, X))$, for suitable neural transforms $\mathcal{T}, \mathcal{T}'$ and a readout NN. In other words, by attending to hyperedges and aggregating across them the Dual HGNN implements learnable functions of high-order statistics (walk counts, diffusion affinities, and role-similarity) that are relevant to controllability/robustness targets and which are generally inaccessible to models restricted to 1-hop adjacency information.

In summary, under mild and natural assumptions on the dependence of the target on multi-step connectivity, K-Hop and K-NN hypergraphs provide a formal instantiation of high-order knowledge; the Dual HGNN is a principled mechanism to transform these high-order statistics into predictive node embeddings and graph-level summaries.

C. The overview of BC_{GAT} Model Network Structure

Table S1 shows the network architecture of the BC_{GAT} model.

D. The experimenal result of Training Spilt in section 4.4.1

TABLE S2
THE MEAN ERROR $\overline{er}$ AND MEAN STANDARD DEVIATION $\overline{\sigma}$ OF THE CONTROLLED ROBUSTNESS CURVES OF NCR-HOK'S PREDICTION FOR DIFFERENT NETWORK TOPOLOGIES ARE REDUCED BY 20%, 40%, 60%, AND 80% FOR THE TRAINING DATASET, RESPECTIVELY.

RA Attack			**<k>=2**	**<k>=5**	**<k>=8**	**<k>=10**
20%	**ER**	$\overline{er}$	0.019	0.015	0.016	0.012
		$\overline{\sigma}$	0.014	0.011	0.010	0.008
	SF	$\overline{er}$	0.020	0.022	0.023	0.023
		$\overline{\sigma}$	0.016	0.016	0.017	0.017
	QSN	$\overline{er}$	0.016	0.018	0.016	0.016
		$\overline{\sigma}$	0.012	0.012	0.011	0.011
	SW	$\overline{er}$	0.016	0.015	0.014	0.011
		$\overline{\sigma}$	0.010	0.011	0.009	0.008
40%	**ER**	$\overline{er}$	0.017	0.015	0.014	0.012
		$\overline{\sigma}$	0.013	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.020	0.022
		$\overline{\sigma}$	0.012	0.015	0.015	0.016
	QSN	$\overline{er}$	0.015	0.016	0.014	0.014
		$\overline{\sigma}$	0.011	0.012	0.010	0.010
	SW	$\overline{er}$	0.013	0.014	0.013	0.011
		$\overline{\sigma}$	0.009	0.010	0.008	0.008
60%	**ER**	$\overline{er}$	0.016	0.015	0.014	0.014
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.020	0.021
		$\overline{\sigma}$	0.012	0.014	0.015	0.015
	QSN	$\overline{er}$	0.015	0.015	0.013	0.013
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.013	0.015	0.012	0.011
		$\overline{\sigma}$	0.009	0.010	0.008	0.007
80%	**ER**	$\overline{er}$	0.016	0.014	0.013	0.011
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.015	0.015
	QSN	$\overline{er}$	0.015	0.015	0.013	0.013
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.012	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.009	0.008	0.007
Full(Raw)	**ER**	$\overline{er}$	0.016	0.015	0.012	0.010
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.015	0.016
	QSN	$\overline{er}$	0.015	0.015	0.013	0.012
		$\overline{\sigma}$	0.011	0.011	0.009	0.008
	SW	$\overline{er}$	0.012	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.009	0.008	0.007

E. The experimenal result of the k-value of the K-Hop and K-NN

1) K-Hop Experiments on the different choice of K: Based on the data in table S3, the choice of K=3 (3-Hop) appears to be a practical and well-balanced decision, representing an excellent trade-off between model performance and computational cost. As shown in the table results:

- Marginal Performance Gains with Higher K: When comparing the results of 3-Hop to 4-Hop, the performance improvements in terms of mean error and standard deviation are minimal to non-existent across all network types. For example, in SW networks with $< k >= 10$, both 3-Hop and 4-Hop achieve the same mean error of 0.010 and standard deviation of 0.007. Since a larger K-value increases the computational complexity of constructing the hypergraph, choosing K=4 offers no significant accuracy benefit over K=3.

- Consistent Improvement Over K=2: While 3-Hop doesn't show a large advantage over 4-Hop, it does offer slight but consistent improvements over 2-Hop in several cases. For instance, in QSN networks with $< k >= 10$, the mean error drops from 0.013 at 2-Hop to 0.012 at 3-Hop. This suggests that moving from K=2 to K=3 captures more valuable high-order information.
- Identical Performance on SF Networks: For SF networks, the performance is identical across all three K-values. In this scenario, selecting the lowest computationally expensive option that still performs well on other network types (like K=3) is the most logical choice.

In summary, K=3 was likely chosen because it hits a sweet spot. It captures more complex network structures than K=2, leading to slightly better performance, while avoiding the increased computational cost of K=4, which provides no meaningful improvement in accuracy.

TABLE S3
THE MEAN ERROR ER AND MEAN STANDARD DEVIATION A OF THE CONTROLLED ROBUSTNESS CURVES OF NCR-HOK'S PREDICTION FOR DIFFERENT NETWORK TOPOLOGIES FOR K-HOP'S K-VALUES OF 2, 3, AND 4 CASES, RESPECTIVELY.

RA Attack			<k>=2	<k>=5	<k>=8	<k>=10
2-Hop	**ER**	$\overline{er}$	0.016	0.014	0.012	0.011
		$\overline{\sigma}$	0.013	0.011	0.010	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.013	0.015	0.016	0.016
	QSN	$\overline{er}$	0.015	0.015	0.013	0.013
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.012	0.015	0.012	0.010
		$\overline{\sigma}$	0.009	0.010	0.009	0.008
3-Hop (Raw)	**ER**	$\overline{er}$	0.016	0.015	0.012	0.010
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.015	0.016
	QSN	$\overline{er}$	0.015	0.015	0.013	0.012
		$\overline{\sigma}$	0.011	0.011	0.009	0.008
	SW	$\overline{er}$	0.012	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.009	0.008	0.007
4-Hop	**ER**	$\overline{er}$	0.016	0.015	0.012	0.011
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.016	0.016
	QSN	$\overline{er}$	0.015	0.014	0.013	0.013
		$\overline{\sigma}$	0.011	0.010	0.009	0.009
	SW	$\overline{er}$	0.013	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.009	0.008	0.007

2) K-NN Experiments on the different choice of K: Based on the data in table S4, the choice of K=10 (10-NN) is a reasonable and justified default because the model's performance shows almost no sensitivity to the K-value in the K-NN hypergraph construction. As shown in the table results:

- Consistent Performance Across All K-Values: When comparing the results for 5-NN, 10-NN, 20-NN, and 30-NN, the mean error and standard deviation are remarkably stable across all network types and average degrees. For example:
  - For SF networks with an average degree of $< k >= 8$, the mean error is consistently 0.021 and the standard deviation is 0.015, regardless of whether K is 5, 10, 20, or 30.
  - For SW networks with $< k >= 10$, both the mean error (0.010) and standard deviation (0.007) are identical across all four K-NN settings.
- No Advantage when Increasing K: Unlike the K-Hop analysis where a larger K sometimes offered marginal benefits, here there is no discernible improvement from increasing K from 10 to 20 or 30. Since a larger K-value increases the density of the hypergraph and thus the computational cost, choosing a higher value offers no practical benefit.

In summary, because the model's prediction accuracy is insensitive to this specific hyperparameter, choosing K=10 serves as a sensible middle-ground. It's large enough to capture local neighborhood structures in the embedding space but not so large as to add unnecessary computational complexity for no performance gain.

TABLE S4
THE MEAN ERROR ER AND MEAN STANDARD DEVIATION A OF THE CONTROLLED ROBUSTNESS CURVES OF NCR-HOK'S PREDICTION FOR DIFFERENT NETWORK TOPOLOGIES FOR K-NN'S K-VALUES OF 5, 10, 20 AND 30 CASES, RESPECTIVELY.

RA Attack			<k>=2	<k>=5	<k>=8	<k>=10
5-NN	**ER**	$\overline{er}$	0.016	0.014	0.012	0.010
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.016	0.015
	QSN	$\overline{er}$	0.015	0.014	0.013	0.012
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.012	0.014	0.011	0.010
		$\overline{\sigma}$	0.010	0.010	0.009	0.007
10-NN (Raw)	**ER**	$\overline{er}$	0.016	0.015	0.012	0.010
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.012	0.015	0.015	0.016
	QSN	$\overline{er}$	0.015	0.015	0.013	0.012
		$\overline{\sigma}$	0.011	0.011	0.009	0.008
	SW	$\overline{er}$	0.012	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.009	0.008	0.007
20-NN	**ER**	$\overline{er}$	0.016	0.014	0.013	0.012
		$\overline{\sigma}$	0.012	0.011	0.010	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.013	0.015	0.016	0.016
	QSN	$\overline{er}$	0.015	0.015	0.013	0.013
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.012	0.015	0.012	0.011
		$\overline{\sigma}$	0.009	0.010	0.008	0.008
30-NN	**ER**	er	0.016	0.015	0.012	0.010
		$\overline{\sigma}$	0.012	0.011	0.009	0.008
	SF	$\overline{er}$	0.016	0.019	0.021	0.021
		$\overline{\sigma}$	0.013	0.015	0.016	0.016
	QSN	$\overline{er}$	0.015	0.014	0.013	0.012
		$\overline{\sigma}$	0.011	0.011	0.010	0.009
	SW	$\overline{er}$	0.013	0.014	0.012	0.010
		$\overline{\sigma}$	0.009	0.010	0.009	0.007

F. Analysis on More Different Attack Scenarios

We supplemented our analysis with Target Degree Based Attacks (TDA) and Target Betweenness Based Attacks (TBA) results. The experimental results, presented in tables S5 and S6, demonstrate that NCR-HoK maintains a significant advantage under both attack scenarios, further highlighting the robustness of our approach.

1) Performance under TDA: As shown in table S5, our model achieves the top rank (#1) in overall average performance for every network type (ER, SF, QSN, and SW) under Target Degree Based Attacks. For instance, on SF networks, NCR-HoK's average error is 0.010, which is substantially lower than the next best model, CRL-SGNN (0.035). Furthermore, our model consistently records the lowest average standard deviation, indicating that its predictions are not only accurate but also highly stable.

2) Performance under TBA: The results under Target Betweenness Based Attacks, detailed in table S6, reinforce this conclusion. Our model again secures the top rank (#1) for overall average error on every network, where it still achieves a competitive first-place rank (#1).

TABLE S5
THE AVERAGE ERRORS $\overline{er}$ AND AVERAGE STANDARD DEVIATIONS $\overline{\sigma}$ OF THE CONTROLLABLE ROBUSTNESS CURVE PREDICTING FOR VARIOUS TYPES OF NETWORKS UNDER THE TDA CONDITION ARE COMPARED AMONG THE NCR-HOK, PCR [18], IPCR [19] MODELS AND CRL-SGNN [20] , ALONG WITH A COMPREHENSIVE RANKING.

			<k>=2	**<k>=5**	**<k>=8**	**<k>=10**	**Average**
ER	**PCR**	$\overline{er}$	0.039	0.040	0.018	0.013	0.028(#3)
		$\overline{\sigma}$	0.017	0.016	0.010	0.008	0.013(#2)
	iPCR	$\overline{er}$	0.053	0.004	0.008	0.006	0.018(#2)
		$\overline{\sigma}$	0.070	0.004	0.007	0.006	0.022(#3)
	CRL-SGNN	$\overline{er}$	0.013	0.070	0.019	0.049	0.038(#4)
		$\overline{\sigma}$	0.014	0.047	0.022	0.051	0.034(#4)
	Our	$\overline{er}$	0.011	0.020	0.012	0.011	**0.014(#1)**
		$\overline{\sigma}$	0.007	0.011	0.007	0.007	**0.008(#1)**
SF	**PCR**	$\overline{er}$	0.073	0.061	0.047	0.042	0.056(#4)
		$\overline{\sigma}$	0.004	0.008	0.009	0.010	0.008(#2)
	iPCR	$\overline{er}$	0.010	0.182	0.012	0.008	0.053(#3)
		$\overline{\sigma}$	0.009	0.198	0.014	0.010	0.058(#4)
	CRL-SGNN	$\overline{er}$	0.026	0.033	0.025	0.057	0.035(#2)
		$\overline{\sigma}$	0.032	0.049	0.035	0.082	0.050(#2)
	Our	$\overline{er}$	0.005	0.010	0.011	0.013	**0.010(#1)**
		$\overline{\sigma}$	0.003	0.005	0.006	0.007	**0.005(#1)**
QSN	**PCR**	$\overline{er}$	0.017	0.021	0.017	0.017	0.018(#2)
		$\overline{\sigma}$	0.007	0.011	0.010	0.010	0.010(#2)
	iPCR	$\overline{er}$	0.009	0.114	0.011	0.008	0.036(#3)
		$\overline{\sigma}$	0.011	0.129	0.013	0.011	0.041(#4)
	CRL-SGNN	$\overline{er}$	0.074	0.059	0.016	0.031	0.045(#4)
		$\overline{\sigma}$	0.026	0.073	0.020	0.037	0.039(#3)
	Our	$\overline{er}$	0.010	0.018	0.014	0.014	**0.014(#1)**
		$\overline{\sigma}$	0.006	0.012	0.010	0.009	**0.009(#1)**
SW	**PCR**	$\overline{er}$	0.010	0.011	0.012	0.011	0.011(#2)
		$\overline{\sigma}$	0.005	0.007	0.008	0.007	0.007(#2)
	iPCR	$\overline{er}$	0.009	0.079	0.011	0.008	0.027(#3)
		$\overline{\sigma}$	0.013	0.093	0.014	0.012	0.033(#3)
	CRL-SGNN	$\overline{er}$	0.023	0.047	0.033	0.029	0.033(#4)
		$\overline{\sigma}$	0.031	0.054	0.039	0.035	0.039(#4)
	Our	$\overline{er}$	0.006	0.012	0.011	0.010	**0.010(#1)**
		$\overline{\sigma}$	0.004	0.007	0.007	0.007	**0.006(#1)**

TABLE S6
THE AVERAGE ERRORS $\overline{er}$ AND AVERAGE STANDARD DEVIATIONS $\overline{\sigma}$ OF THE CONTROLLABLE ROBUSTNESS CURVE PREDICTING FOR VARIOUS TYPES OF NETWORKS UNDER THE TBA CONDITION ARE COMPARED AMONG THE NCR-HOK, PCR [18], IPCR [19] MODELS AND CRL-SGNN [20] , ALONG WITH A COMPREHENSIVE RANKING.

			<k>=2	**<k>=5**	**<k>=8**	**<k>=10**	**Average**
ER	**PCR**	$\overline{er}$	0.047	0.033	0.018	0.016	0.029(#3)
		$\overline{\sigma}$	0.017	0.015	0.012	0.010	0.014(#2)
	iPCR	$\overline{er}$	0.080	0.013	0.011	0.013	0.029(#3)
		$\overline{\sigma}$	0.086	0.009	0.011	0.013	0.030(#3)
	CRL-SGNN	$\overline{er}$	0.014	0.041	0.015	0.038	0.027(#2)
		$\overline{\sigma}$	0.015	0.052	0.014	0.038	0.030(#3)
	Our	$\overline{er}$	0.017	0.014	0.016	0.012	**0.015(#1)**
		$\overline{\sigma}$	0.012	0.010	0.010	0.009	**0.010(#1)**
SF	**PCR**	$\overline{er}$	0.110	0.069	0.050	0.044	0.068(#4)
		$\overline{\sigma}$	0.014	0.017	0.015	0.015	0.015(#2)
	iPCR	$\overline{er}$	0.014	0.180	0.013	0.013	0.055(#3)
		$\overline{\sigma}$	0.012	0.204	0.012	0.011	0.060(#4)
	CRL-SGNN	$\overline{er}$	0.038	0.027	0.022	0.040	0.032(#2)
		$\overline{\sigma}$	0.042	0.030	0.021	0.042	0.034(#3)
	Our	$\overline{er}$	0.010	0.014	0.014	0.013	**0.013(#1)**
		$\overline{\sigma}$	0.008	0.010	0.010	0.010	**0.010(#1)**
QSN	**PCR**	$\overline{er}$	0.027	0.023	0.021	0.019	0.023(#2)
		$\overline{\sigma}$	0.012	0.012	0.013	0.012	0.012(#2)
	iPCR	$\overline{er}$	0.017	0.127	0.014	0.014	0.043(#4)
		$\overline{\sigma}$	0.013	0.137	0.012	0.013	0.044(#4)
	CRL-SGNN	$\overline{er}$	0.038	0.043	0.020	0.036	0.034(#3)
		$\overline{\sigma}$	0.041	0.051	0.025	0.042	0.040(#3)
	Our	$\overline{er}$	0.014	0.015	0.014	0.013	**0.014(#1)**
		$\overline{\sigma}$	0.008	0.009	0.009	0.009	**0.009(#1)**
SW	**PCR**	$\overline{er}$	0.014	0.021	0.018	0.016	0.017(#2)
		$\overline{\sigma}$	0.006	0.013	0.012	0.010	**0.010(#1)**
	iPCR	$\overline{er}$	0.012	0.096	0.016	0.013	0.034(#4)
		$\overline{\sigma}$	0.013	0.099	0.015	0.013	0.035(#4)
	CRL-SGNN	$\overline{er}$	0.015	0.037	0.034	0.040	0.032(#3)
		$\overline{\sigma}$	0.015	0.042	0.037	0.047	0.033(#3)
	Our	$\overline{er}$	0.009	0.013	0.014	0.013	**0.012(#1)**
		$\overline{\sigma}$	0.004	0.013	0.014	0.013	0.011(#2)

REFERENCES

[1] K. Maile, E. Lecarpentier, H. Luga, and D. G. Wilson, "Darts-prime: Regularization and scheduling improve constrained optimization in differentiable nas," *arXiv preprint arXiv:2106.11655*, 2021.

[2] H. Yue, P. Hong, and H. Liu, "Graph–graph similarity network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 7, pp. 9136–9146, 2022.

[3] H. Wang, J. Yu, X. Wang, C. Chen, W. Zhang, and X. Lin, "Neural similarity search on supergraph containment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 281–295, 2023.

[4] M. Isallari and I. Rekik, "Brain graph super-resolution using adversarial graph neural network with application to functional brain connectivity," *Medical Image Analysis*, vol. 71, p. 102084, 2021.

[5] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph Neural Networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.

[6] M. Lu, Z. Xiao, H. Li, Y. Zhang, and N. N. Xiong, "Feature pyramid-based graph convolutional neural network for graph classification," *Journal of Systems Architecture*, vol. 128, p. 102562, 2022.

[7] X. Zhou and H. Wu, "schiclassifier: a deep learning framework for cell type prediction by fusing multiple feature sets from single-cell hi-c data," *Briefings in Bioinformatics*, vol. 26, no. 1, 2024.

[8] X. Yang, K. K. Mann, H. Wu, and J. Ding, "sccross: a deep generative model for unifying single-cell multi-omics with seamless integration, cross-modal generation, and in silico exploration," *Genome Biology*, vol. 25, no. 1, p. 198, 2024.

[9] S. Zhou, J. Zhang, W. Zuo, and C. C. Loy, "Cross-scale internal graph neural network for image super-resolution," *Advances in neural information processing systems*, vol. 33, pp. 3499–3509, 2020.

[10] Z. Xia, W. Zhang, and Z. Weng, "Social recommendation system based on hypergraph attention network," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 7716214, 2021.

[11] B. Yu, C. Xie, H. Cai, H. Duan, and P. Tang, "Meta-path and hypergraph fused distillation framework for heterogeneous information networks embedding," *Information Sciences*, vol. 667, p. 120453, 2024.

[12] J. Zhu, A. Reganti, E. W. Huang, C. Dickens, N. Rao, K. Subbian, and D. Koutra, "Simplifying distributed neural network training on massive graphs: Randomized partitions improve model aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol. 19, no. 1, pp. 1–26, 2025.

[13] E. Chien, C. Pan, J. Peng, and O. Milenkovic, "You are allset: A multiset function framework for hypergraph neural networks," *arXiv preprint arXiv:2106.13264*, 2021.

[14] X. Zhi, "A review of hypergraph neural networks." *EAI Endorsed Transactions on e-Learning*, vol. 10, 2024.

[15] L. Bai, F. Hu, C. Tang, Z. Mei, and C. Liu, "Hyperbolic multi-channel hypergraph convolutional neural network based on multilayer hypergraph," *Scientific Reports*, vol. 15, no. 1, p. 24606, 2025.

[16] S. Imai and A. Inokuchi, "Efficient supergraph search using graph coding," *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 1, pp. 130–141, 2020.

[17] Z. Chang, Y. Zhang, and H. C. Chen, "Neural network graph similarity

computation based on graph fusion," *arXiv preprint arXiv:2502.18291*, 2025.

[18] Y. Lou, Y. He, L. Wang, and G. Chen, "Predicting Network Controllability Robustness: A Convolutional Neural Network Approach," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 4052–4063, 2020.

[19] Y. Lou, Y. He, L. Wang, K. F. Tsang, and G. Chen, "Knowledge-Based Prediction of Network Controllability Robustness," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5739–5750, 2021.

[20] Y. Zhang, J. Ding, and X. Li, "Network controllability robustness learning via spatial graph neural networks," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 5, pp. 4045–4058, 2024.