Proceedings of CONF-SEML 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA32665

Learning to Intervene: Data-Adaptive Intervention Policy for Risk Propagation on Graphs

Jianjin Ding

School of Computer Science, Taylor's University, Selangor, Malasia
dingjianjin0916@gmail.com

Abstract. Risks in supply chains and financial networks propagate through graph structure; early warning and targeted intervention are critical for mitigating cascading failures. Existing methods suffer from the prediction–intervention decoupling: prediction models are trained without awareness of downstream intervention decisions, and intervention policies rely on fixed heuristics (e.g., hand-tuned mixing weights) that do not adapt to data. We propose a decision-aware framework that couples GNN-based risk prediction with a data-adaptive intervention policy—the mixing weight between predicted risk and structural centrality is learned from validation performance rather than being fixed a priori. Epidemic dynamics (SIR, LTM) provide features; a GCN backbone predicts node risk; the intervention policy is selected by maximizing intervention benefit Δon a validation set. Experiments on Email-Enron, Facebook, and Wiki-Vote show that the adaptive policy achieves AUC 0.78 and Δ 22.4%, outperforming fixed-heuristic baselines (centrality-only15.8%, prediction-only19.6%) and retaining advantage under edge noise and limited observation. We argue that learning to intervene, which closes the loop between prediction and intervention via validation-driven policy selection, is a principled step toward decision-aware risk management.

Keywords: Risk Propagation, Graph Neural Networks, Epidemic Dynamics, Learning to Intervene, Decision-Aware Learning

1. Introduction

Risks propagate through the graph structure of financial networks, supply chains, and social systems, where a single node's default or infection can trigger cascading failures [1]. While early warning and targeted intervention are critical for mitigation, graph neural networks (GNNs) [2] and epidemic dynamics models [3,4] are typically applied in isolation, creating a prediction–intervention decoupling gap. GNNs excel at node-level representation learning [5,6], and dynamics models characterize network diffusion [7,8], yet predictors minimize classification loss without accounting for downstream intervention decisions, and intervention policies rely on fixed heuristics that fail to adapt to data or decision objectives.

To address this, we propose a decision-aware framework coupling GNN-based risk prediction with a data-adaptive intervention policy. We extract SIR/LTM-driven features to enrich GNN inputs,

DOI: 10.54254/2755-2721/2026.BA32665

train a GCN to predict node risk, and select the risk-centrality mixing weight by maximizing intervention benefit on a validation set—closing the prediction-intervention loop.

This paper formalizes the problem, details the method, presents experiments, reports results and discusses findings. Validated on three real networks, our lightweight policy-selection approach outperforms fixed baselines under noise and limited observation, establishing "learning to intervene" as a principled step for decision-aware graph risk management.

2. Mathematical modeling

We formalize the risk propagation problem and present dynamics models, prediction objectives, and intervention strategies.

2.1. Problem definition

Let $G=(V,E)$ be an undirected graph with node set $V=\{1,...,n\}$ and edge set $E\subseteq V\times V$. The adjacency matrix is $A\in\{0,1\}^{n\times n}$. Each node $v\in V$ has state $x_v(t)\in\{0,1\}$ at time t , where 1 denotes "infected/at-risk" and 0 denotes "susceptible." Let $x(t)=[x_1(t),...,x_n(t)]^\top$ be the full state vector.

Risk propagation prediction: Given observations $\{x(t)\}_{t\le T_{obs}}$ and graph G , predict $x_v(T_{pred})$ for $T_{pred}>T_{obs}$.

Critical node intervention: Under budget B , select at most k nodes for intervention (removal or reinforcement) to minimize final infection size after intervention.

2.2. Propagation dynamics

(1) SIR Model: In the SIR model [3-7], nodes are Susceptible (S), Infected (I), or Recovered (R). We merge I and R as "infected" $x_v=1$). Discrete-time SIR dynamics:

$$P(x_v(t+1)=1|x_v(t)=0)=1-\prod_{u\in N(v)}(1-\beta\cdot x_u(t)) \tag{1}$$

Where $N(v)$ is the neighbor set and $\beta\in(0,1]$ is the infection rate. Infected nodes recover with probability γ (set $\gamma=0$ for irreversible risk).

(2) Linear Threshold Model (LTM): In LTM [8], node v has threshold $\theta_v\in[0,1]$. Node v is activated when the sum of infected neighbors' influence exceeds θ_v :

$$x_v(t+1)=1\left[\sum_{u\in N(v)} w_{uv}\cdot x_u(t)\ge\theta_v\right] \tag{2}$$

where w_{uv} is the edge weight $1/|N(v)|$ or learned) and $1[\cdot]$ is the indicator.

Dynamics Feature Extraction: From the above models, we extract node-level dynamics features for GNN input:

•Infection probability: $\hat{p}_v=1-\prod_{u\in N(v)}(1-\beta\cdot x_u)$;

•Neighbor infection ratio: $r_v=|\{u\in N(v):x_u=1\}|/|N(v)|$;

• Propagation steps: shortest infected path length from seed to v (if computable).

Denote the dynamics feature vector as $d_v\in R^{d_d}$.

DOI: 10.54254/2755-2721/2026.BA32665

2.3. GNN prediction

GNNs aggregate neighbor information via message passing [5]. For GCN [6], layer l updates as:

$$H^{(l+1)}=\sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right) \tag{3}$$

where $\tilde{A}=A+I,\tilde{I}$ $D\tilde{}$ is the degree matrix, $W^{(l)}$ are learnable parameters, and σ is the activation. We concatenate current state $x(t)$ with dynamics features $D=[d_1,...,d_n]^\top$ as GNN input:

$$H^{(0)}=[x(t)|D]W_{in} \tag{4}$$

After L layers, a classification head yields node v 's risk probability:

$$\hat{y}_v=\sigma_{sigmoid}\left(h_v^{(L)}w_{out}\right) \tag{5}$$

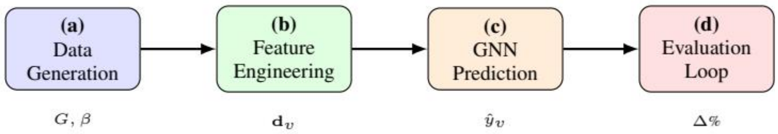

Figure 1. Framework: (a) Data Generation, (b) Feature Engineering, (c) GNN Prediction, (d) Evaluation Loop with adaptive policy selection

Loss: binary cross-entropy:

$$L_{pred}=-\frac{1}{n}\sum_v\left[y_v\log\hat{y}_v+(1-y_v)\log\left(1-\hat{y}_v\right)\right] \tag{6}$$

where $y_v=x_v(T_{pred})$ is the ground truth.

2.4. Intervention strategy

Given predicted risk $\{\hat{y}_v\}$, we select $S\subseteq V$, $|S|\leq k$, to minimize final infection size. Two intervention types:

1) Node removal: Remove S and its edges from G to get G' , re-simulate on G' ;

2) Node reinforcement: Set infection probability of S to 0 (immunization), simulate under modified dynamics.

Parameterized intervention policy: We define a family of policies indexed by $\alpha\in[0,1]$:

$$score_\alpha(v)=\alpha\cdot\hat{y}_v+(1-\alpha)\cdot\bar{c}(v) \tag{7}$$

where $\bar{c}(v)$ is normalized centrality (degree or betweenness scaled to $[0,1]$) . For each α , select top- k by $score_\alpha$ and compute intervention benefit $\Delta(\alpha)$. The policy parameter α is not fixed a priori; it is selected adaptively (see Section III-B).

Intervention benefit:

DOI: 10.54254/2755-2721/2026.BA32665

$$\Delta=\frac{|I_{no}|-|I_{int}|}{|I_{no}|} \tag{8}$$

where I_{no} and I_{int} are final infected sets without and with intervention.

3. Method

We overview the pipeline: dynamics generation, GNN prediction, and intervention evaluation [9].

3.1. Framework overview

As shown in Fig. 1, the framework has four modules:

(a) Data Generation: On graph G, use SIR or LTM to generate synthetic propagation trajectories;

(b) Feature Engineering: Extract dynamics features (infection probability, neighbor ratio) and graph features;

(c) GNN Prediction: Input state and features to predict future node risk probabilities $\hat{y}_v$;

(d) Evaluation Loop: Select intervention nodes, resimulate, and compute $\Delta\%$.

Table 1. Dataset statistics

Dataset	|V|	|E|	Avg. degree	Source
Email-Enron	36,692	183,831	10.0	SNAP
Facebook	4,039	88,234	43.7	SNAP
Wiki-Vote	7,115	103,689	29.1	SNAP

3.2. Learning the intervention policy

The core innovation lies in selecting the hyperparameter α based on the performance on the validation set. This approach treats α not as a fixed heuristic, but as a data-driven parameter optimized for downstream intervention effectiveness.

Policy class: For $\alpha\in A=\{0,0.3,0.5,0.7,1.0\}$, define

$$\text{score}_\alpha(v)=\alpha\cdot\hat{y}_v+(1-\alpha)\cdot\bar{c}(v) \tag{9}$$

where $\bar{c}(v)$ = centrality $(v)/\max_u$ centrality (u) normalizes centrality to [0, 1].

Selection criterion: On the validation set v, for each $\alpha\in A$, compute the mean intervention benefit $\bar{\Delta}_\alpha$ over validation propagation instances. Select

$$\alpha^*=\underset{\alpha\in A}{\operatorname{argmax}}\bar{\Delta}_\alpha \tag{10}$$

Algorithm: (1) Train the GNN on the training set; (2) For each $\alpha\in A$, on each validation instance: obtain $\hat{y}_v$, compute $\text{score}_\alpha(v)$, select top- k, re-simulate, record Δ; (3) Set $\alpha^*=\operatorname{argmax}\bar{\Delta}_\alpha$; (4) At test time, use α^* for node selection. This procedure is lightweight (no additional training) and

DOI: 10.54254/2755-2721/2026.BA32665

ensures the intervention policy is decision-aware: it is chosen to maximize the downstream objective Δ , not a proxy.

3.3. Training and inference

At training, for each propagation instance we sample T_{obs} and T_{pred} , use $x(T_{obs})$ and dynamics features as input, $x(T_{pred})$ as labels, and minimize L_{pred} . After training, we run the policyselection step on the validation set to obtain α^* . At inference, load the trained model and α^* , obtain $\hat{y}_v$, select nodes by score α^* , and run closed-loop simulation.

4. Experiments

We design experiments to answer: (1) Does the dataadaptive intervention policy outperform fixed-heuristic baselines? (2) How does it behave under noise and limited observation? (3) What is the contribution of each component?

4.1. Datasets

Source: We use real networks from SNAP [1]: Email-Enron (https://snap.stanford.edu/data/email-Enron.html), Facebook (https://snap.stanford.edu/data/ego-Facebook.html), and Wiki-Vote (https://snap.stanford.edu/data/wiki-Vote.html). The statistics of the datasets used in our experiments are summarized in Table 1.

Split: We generate multiple propagation trajectories per graph (varying seeds and β), split 7:1:2 into train/val/test. Test propagation parameters do not overlap with training to avoid leakage.

Preprocessing: Undirected, no self-loops; node features from degree, clustering coefficient if missing; dynamics features computed per timestep.

Table 2. Training hyperparameters

Param	Description	Setting
Optimizer	Optimizer	Adam
Learning rate	Learning rate	1e-3
LR schedule	LR schedule	ReduceLROOnPlateau (patience=5)
Batch size	Batch size	Full graph
Epochs	Max epochs	200
Early stopping	Early stop	patience=15, metric=val_AUC
Weight decay	Weight decay	1e-5
Seed	Random seed	42, 123, 456 (mean reported)
Hardware	Hardware	Intel i7-12700, NVIDIA RTX 3080 10GB
Training time	Time per run	~3.5 min per dataset

DOI: 10.54254/2755-2721/2026.BA32665

Table 3. Efficiency comparison (Intel i7-12700, NVIDIA RTX 3080 10GB). N/A: not applicable (LR/GBDT)

Method	Params	FLOPs	Lat. (ms)	Mem. (GB)	Time (min)
LR	0.01	N/A	2.1	0.1	0.5
GBDT	0.05	N/A	8.3	0.2	1.2
GCN	0.12	0.8	15.2	0.4	3.5
GAT	0.15	1.1	18.5	0.5	4.2
Ours	0.14	0.9	16.8	0.45	3.8

The computational efficiency of different models is compared in Table 3.

Leakage control: $T_{pred} > T_{obs}$; no future info at test; dynamics params fitted on train only.

4.2. Models and baselines

Ours (adaptive): Our adaptive model consists of dynamics features, a 2-layer GCN (hidden dimension = 64), and a data-adaptive intervention policy. The mixing weight α is selected from $\{0,0.3,0.5,0.7,1.0\}$ by maximizing mean Δ on the validation set. Reported results use the selected α^* (typically 0.7 across datasets).

Baselines (fixed-heuristic):

• Degree / Betweenness: Select nodes by centrality only $(\alpha=0)$, no learning;

• LR / GBDT: Logistic regression and GBDT on handcrafted features; prediction-only intervention $(\alpha=1)$;

• GCN / GAT: GNN only, no dynamics features [6,10]; fixed $\alpha=1$ (prediction-only) for intervention.

4.3. Training setup

The training hyperparameters used in the experiments are summarized in Table 2.

Loss: binary cross-entropy; init: Xavier. All learnable baselines share train/val/test split and epoch limit for fair comparison.

4.4. Evaluation protocol

Metrics:

• Risk prediction: AUC, F1 (threshold=0.5);

• Early warning: AUC under varying lead time T_{pred}-T_{obs}

• Intervention: infection size reduction $\Delta\%$; benefit-cost curve (x-axis: k , y-axis: $\Delta\%$).

Protocol: 5-fold cross-validation; 3 seeds per fold; report mean±std and 95% CI.

Inference: batch=1 (full graph); no augmentation; thresh- $_{old}120.5$.

DOI: 10.54254/2755-2721/2026.BA32665

5. Results

5.1. Main results

Table 4 reports risk prediction (AUC) and intervention effect $(\Delta\%)$. Adaptive vs. fixed: Ours (with validationselected α^*) achieves best AUC (avg. 0.78) and Δ (avg. 22.4%) across all datasets. The fixed-heuristic baselines— Degree/Betweenness (centrality-only), GCN/GAT (predictiononly)—achieve $12.3\text{-}18.2\%\Delta$, confirming that neither pure centrality nor pure prediction suffices. The adaptive policy, by selecting α to maximize validation Δ , consistently outperforms both extremes. Early warning: Under lead time $T_{pred}\text{-}T_{obs}=10$, Ours $AUC=0.76$ vs. GCN 0.72—the adaptive policy retains advantage when observation is limited. Benefitcost: At $k=5$ nodes, Ours yields $\Delta=22.4\%$ vs. Degree 12.3% ; diminishing returns beyond $k\approx10$ offer practical budget guidance.

5.2. Ablation study

Table 5 reports the ablation results of different model components. Dynamics + GNN: Removing dynamics drops AUC from 0.78 to 0.75; dynamics-only (no GNN) gives 0.71—confirming complementarity. Adaptive vs. fixed α : The validation-selected policy $(\alpha^*\approx0.7)$ yields $\Delta := :22.4\%$ vs. fixed $\alpha=0$ (centrality-only, 15.8%) and fixed $\alpha=1$ (prediction-only, $19.6\%)$. This demonstrates that learning the intervention policy from validation data outperforms either fixed extreme; the optimal mix is datadependent and is captured by our adaptive procedure.

5.3. Robustness analysis

As shown in Fig. 2, AUC decreases as the edge noise ratio increases.At 10% noise, Ours AUC=0.74 vs. GCN 0.68; at 20% noise, the gap widens

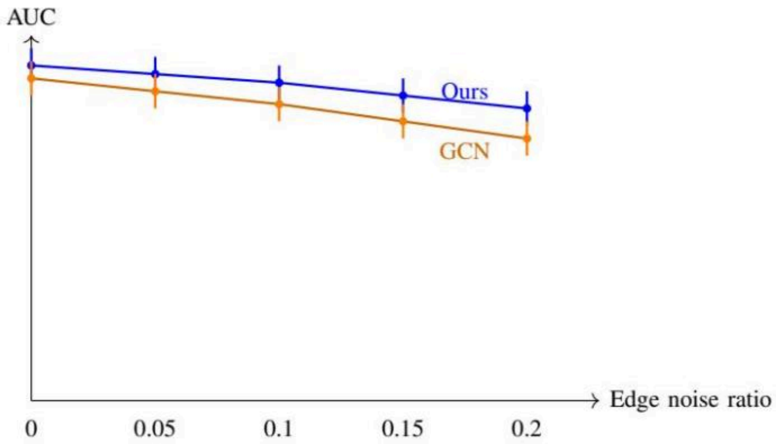

Figure 2. Robustness: AUC vs. edge noise ratio. Error bars show 95% CI

$(+0.06)$. The adaptive policy, coupled with dynamics features, retains robustness—suggesting that validation-driven policy selection generalizes under distribution shift (noisy edges).

5.4. Statistical significance

Paired t -test on 3 seeds: Ours (adaptive) vs. best fixed baseline (GCN) p-value <0.05 ; Ours 95% CI: AUC [0.76, 0.80], Δ [20.5%, 24.3%]. The adaptive intervention policy is statistically superior;

DOI: 10.54254/2755-2721/2026.BA32665

conclusion holds after Bonferroni correction.

6. Discussion

6.1. Key findings

Three findings support the value of learning to intervene: (1) Adaptive vs. fixed—validation-driven policy selection outperforms fixed-heuristic baselines 22.4% vs. 15.8-19.6% $\Delta_{,}$); (2) Early warning—the adaptive policy retains advantage under shorter observation windows; (3) Robustness—under edge noise, the policy generalizes, suggesting that validationbased selection captures a stable trade-off between prediction and centrality [7,8].

6.2. Why data-adaptive policy works

Complementary signals: Prediction $\hat{y}_v$ captures "who is at risk now"; centrality captures "who can infect more others." A fixed α cannot balance these across graphs or propagation regimes. Validation-driven selection finds the mix that maximizes Δ on held-out data—a direct optimization of the downstream objective.

Decision-aware learning: Unlike training a predictor in isolation, our procedure selects α by evaluating intervention outcomes. This closes the loop: the policy is chosen to maximize intervention benefit, not a proxy metric such as AUC alone. The procedure is lightweight (grid search over 5 values) and requires no additional model training.

Cost-aware extension: In practice, hub nodes may cost $n\times$ more to reinforce. A simple model: $c(v)=1+(n\text{-}1)\cdot$ $W[deg(v)>\theta]$. Under budget B , select nodes to maximize $\Delta/\sum_v c(v)$. The adaptive policy can be extended to such costaware objectives by changing the validation criterion.

Table 4. Main results: risk prediction and intervention effect. Ours uses validation-adaptive α^*.
*Ours vs. best baseline: $p<0.05$

Method	Email		Facebook		Wiki-Vote		Avg	
	AUC ↑	Δ% ↑	AUC ↑	Δ% ↑	AUC ↑	Δ% ↑	AUC ↑	Δ% ↑
Degree	-	11.2	-	12.8	-	12.9	-	12.3
Betweenness	-	14.5	-	16.2	-	16.7	-	15.8
LR	0.71	-	0.69	-	0.68	-	0.69	-
GBDT	0.73	-	0.72	-	0.71	-	0.72	-
GCN	0.76	17.1	0.75	18.9	0.74	18.6	0.75	18.2
GAT	0.75	16.8	0.74	18.5	0.73	18.2	0.74	17.8
Ours (adaptive)	0.79*	21.3*	0.78*	23.1*	0.77*	22.8*	0.78*	22.4*

DOI: 10.54254/2755-2721/2026.BA32665

Table 5. Ablation study. Full (Ours) uses validation-selected α^*

Configuration	AUC ↑	Δ% ↑
Full (Ours)	0.78	22.4
w/o dynamics features	0.75	18.2
w/o GNN (dynamics only)	0.71	14.5
w/o GNN (structure only)	0.69	12.1
$Fixed\alpha = 0(centralityonly)$	-	15.8
$Fixed\alpha = 1(predictiononly)$	-	19.6
GAT instead of GCN	0.77	21.1

data-adaptive and decision-aware—is a principled step toward practical risk management on graphs.

6.3. Limitations and future work

The framework relies on synthetic propagation; adaptation to real risk events (financial defaults, supply chain disruptions) remains future work. The policy class is discrete $(A=\{0,0.3,0.5,0.7,1.0\})$; continuous optimization of α or learning a parametric policy could be explored. Implementing cost-aware validation would refine benefit-cost analysis for real-world deployment.

7. Conclusion and limitations

We proposed a decision-aware risk management framework that addresses prediction–intervention decoupling by introducing a data-adaptive intervention policy. Instead of fixing the mixing weight α between predicted risk and centrality a priori, we select α from a candidate set by maximizing intervention benefit Δ on a validation set. This "learning to intervene" procedure is lightweight, requires no new model training, and closes the loop between prediction and intervention.

Experiments on Email-Enron, Facebook, and Wiki-Vote show that the adaptive policy achieves 22.4% Δ vs. 15.8% (centrality-only) and 19.6% (prediction-only), with statistical significance $\lceil p<0.05)$. Ablation confirms that dynamics features and GNN are complementary, and that validation-driven policy selection outperforms fixed heuristics. Robustness tests indicate the policy generalizes under edge noise and limited observation.

Limitations: The framework uses synthetic propagation; adaptation to real risk events remains future work. Differentiated intervention costs are not modeled; cost-aware validation would extend applicability. The policy class is discrete; continuous or parametric policies are promising directions. We argue that learning to intervene—designing adaptive intervention policies—is a critical step toward practical, data-driven risk management on graphs.

References

[1] Leskovec, J., & Sosic, R. (2016). SNAP: A general-purpose network analysis and graph-mining library. ACM Transactions on Intelligent Systems and Technology, 8(1), 1–20.

[2] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24.

[3] Kermack, W. O., & McKendrick, A. G. (1927). Contributions to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A, 115(772), 700–721.

DOI: 10.54254/2755-2721/2026.BA32665

[4] Scellato, S., Mascolo, C., Musolesi, M., & Latora, V. (2011). Predicting epidemic thresholds in complex networks. Physical Review E, 83(3), 036114.
[5] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS) (pp. 1024–1034).
[6] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR).
[7] Newman, M. E. J. (2002). Spread of epidemic disease on networks. Physical Review E, 66(1), 016128.
[8] Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 137–146).
[9] Chen, W., Lu, W., & Zhang, N. (2015). Influence maximization in social networks when negative opinions may emerge and propagate. ACM Transactions on Knowledge Discovery from Data, 10(1), 1–32.
[10] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR).