Journal of Mathematical Finance and Risk Management
ISSN: 3080-9983 (Print), ISSN: 3080-9991 (Online) | Vol. 1, No. 3, 2026

Directional Exposure in Spatio-Temporal Supply-Chain Risk Prediction: Graph Networks Versus a Strong Linear Baseline

Mina Cruz *, Irene Hall, Xinyi Desai

Department of Computer Sciences, University of Wisconsin–Madison, WI, USA

* **Corresponding author**: Mina Cruz (Email: mina.cruz592@wisc.edu)

Abstract: Directed supplier exposures may improve disruption forecasting, but whether nonlinear graph learning adds value beyond interpretable event-history models is uncertain. We implement a two-round graph embedding, aggregation and update baseline and evaluate a directional temporal exposure graph network (DTEGN) whose logit includes lagged one- and two-hop observed incidents. Experiments use exclusively synthetic node attributes, supplier graphs and incident labels from a fully specified four-tier simulator; no operational failures are presented as real. On one chronological held-out test window, DTEGN reaches mean AUPRC 0.237 ± 0.004 across three initializations, compared with 0.194 ± 0.014 for two-round message passing. In a stronger comparison, exposure-augmented logistic regression scores AUPRC 0.279 on that window. After fitting each model on the original simulated world, we evaluate unchanged parameters on 12 independently generated worlds. Mean AUPRCs are 0.369 for DTEGN, 0.321 for two-round message passing and 0.413 for exposure-augmented logistic regression. Thus explicit directed incident exposure is a useful inductive bias, whereas gate complexity is not justified by this additive-hazard simulator. The study contributes an auditable technical implementation, transfer protocol, ablations and negative as well as positive results; real-world validity remains untested.

Keywords: Supply-chain disruptions, directed graph neural networks, lagged exposure, linear baselines, synthetic transfer, risk prediction.

1. Introduction

Interconnected supplier relationships transform a local disruption into a network-level operational exposure. Evidence from multi-tier case studies distinguishes the nature of an initiating event, structural dependence, and managerial decisions as propagation drivers [1]. Outbreak simulations demonstrate that shocks may persist and travel through network stages [2], while the X-network perspective frames the supply network as a reconfigurable system rather than a static procurement tree [3]. A useful prediction model must therefore account for direction (who supplies whom), propagation delay, local vulnerability and changing observations. The corresponding modeling problem is harder than classifying a firm from its current financial or operational indicators.

Data-driven studies have predicted first-tier supplier disruptions from engineered operational attributes [4] and examined the interpretability–performance trade-off in delivery-delay risk prediction [5]. Network-resilience analyses demonstrate the relevance of structural characteristics under propagation [6], whereas supply-network extraction from text [7] and hidden-link prediction with graph neural networks [8] show how difficult complete visibility can be. These studies motivate a design that retains an observable supplier graph when available but treats the timing and direction of indirect exposure as distinct inductive biases. A graph network that averages both downstream demand and upstream incident signals without distinction could conflate causal pathways.

This paper contributes a controlled assessment of when a directed temporal exposure graph network (DTEGN) is warranted. Its defining feature is a learnable additive path for observed recent incidents propagated from suppliers to customers over one and two network hops; a separate multi-hop representation pathway incorporates arbitrary node covariates. An auxiliary sigmoid gate can attenuate learned messages when partner relevance varies. Rather than attributing improvements to "graph learning" as a whole, we isolate directional transmission, two-hop reach, temporal history, explicit incident exposure and gating in controlled ablations. A strong exposure-augmented generalized linear model tests whether additional representation learning is necessary. The graph reference is a two-round message-passing network that follows the embedding–aggregation–update workflow established for supply-chain GNNs, so the value of directed exposure is measured against a representative GNN rather than only against graph-ignorant models.

The benchmark has an important scope: public firm-level time series that jointly contain supplier links, observed precursors and subsequent cascade labels are rarely available under open licenses. We therefore evaluate on a completely synthetic, explicitly documented generator. The objective is mechanistic identifiability and auditability, not an estimate of accuracy for deployed commercial supply chains. To avoid temporal leakage, the last observation at decision time t is used only to predict the incident at $t+1$; normalization is fitted on training observations alone.

2. Related Work and Design Motivation

2.1. Relational and Multi-Hop Modeling

The broader graph-learning literature identifies recurring design choices: normalization, aggregation, attention and

update functions [10]. Graph convolution mixes features from local neighborhoods [11], GraphSAGE constructs inductive embeddings by sampling and aggregating neighbors [12], and graph attention uses learned neighbor weights rather than uniform mixing [13]. The general message-passing formalism separates message construction, aggregation and state update [14]. These operations are useful for supply-chain tiers, but depth alone is not a substitute for specifying edge direction: reverse aggregation answers a different operational question. Representational studies also establish limitations of particular neighborhood aggregators, cautioning against assuming that two hops capture all relevant network patterns [15]. For supply networks, the review of Liu, Li and Wang [9] organizes these operations into an embedding–aggregation–update workflow—node embeddings initialized from entity attributes, successive neighborhood aggregation and representation updates—through which multi-tier dependencies, disruption cascades and indirect supplier exposure can be modeled. We adopt this workflow as the graph baseline of the study and ask what explicit directed incident exposure adds to it.

2.2. Time, Topology and Direction

Spatio-temporal convolution has achieved effective graph-conditioned forecasting in transportation [16]. Diffusion convolutional recurrent networks explicitly model directed walks and temporal state [17]; Graph WaveNet learns potentially missing spatial dependence alongside dilated temporal convolutions [18]. EvolveGCN changes model parameters across graph snapshots when topology evolves [19]. These contributions concern traffic or general dynamic graphs, not observed firm-level supply failures. We borrow the general need for temporal conditioning and directed propagation, while selecting a smaller architecture whose dependency path can be inspected without claiming architectural equivalence to those systems. A digital supply-chain-twin perspective further supports evaluating explicit disruption scenarios as controlled what-if systems [20].

The gap addressed here is operational rather than a new graph-convolution primitive. In short-term incident forecasting, a past incident at an upstream supplier is qualitatively different from a latent embedding transmitted through a customer link. Encoding an event-transfer term explicitly, in addition to multi-hop representations, makes the model responsive to known incidence history and allows the transfer contribution to be disabled, reversed or truncated in a controlled experiment. Because the benchmark generator exposes that very mechanism, the test interrogates whether this bias is recovered; future observational benchmarks will be needed to determine when the same constraint helps under incomplete or disputed supplier graphs.

3. Problem Formulation

Let $G=(V, E)$ be a directed supplier-dependency graph, with an edge $j \to i$ if supplier j serves dependent company i. At each synthetic day t, node i has observable operational features $x_{i,t} \in R^6$ and an incident indicator $y_{i,t} \in \{0,1\}$. We estimate $p_{i,t+1}=P(y_{i,t+1}=1 \mid x_{1:t}, G)$ from the latest $L=5$ days, without observing any target-day feature. The risk output is a sigmoid probability (without assuming calibration), evaluated as a ranking and a proper probability score; no specific intervention action is inferred. All entities are shared across splits, so the task measures forecasting later time points on a known synthetic network rather than induction to new suppliers.

Define the directed matrix A by $A_{ij}>0$ only if j supplies i. Random nonnegative dependency weights are divided by $\max(1, \text{the row sum})$, so A is row-substochastic. $A v$ gives the one-hop upstream aggregate arriving at downstream nodes; $A^2 v$ is the two-hop aggregate. Because reversing A changes incoming suppliers into outgoing dependents, A and A^T cannot be interchanged in a cascade-risk question. A zero in A is assumed to mean no modeled link, not verified absence of a real business relationship.

4. Directed Temporal Exposure Network

4.1. Two-round Graph Baseline

Our graph baseline instantiates the supply-chain GNN workflow of Liu et al. [9] with two rounds of message passing: node embeddings are initialized from local attributes, aggregated once over the supplier neighborhood and updated, then aggregated and updated a second time. Specifically, $h^0_i=\mathrm{ReLU}(W_0\ \mathrm{vec}(x_{i,t-4:t})+b_0)$ and $h^k_i=\mathrm{ReLU}(W_k[h^{k-1}_i;\ \Sigma_j A_{ij}h^{k-1}_j]+b_k)$ for $k=1,2$. This GraphSAGE-style concatenation update [12] is applied over directed supplier neighborhoods, and a linear classifier maps h^2 to the event logit. The baseline receives the same five-day feature window, optimizer and training budget as DTEGN (Section V-D), so differences between the two models isolate the explicit directed exposure path.

4.2. Proposed Exposure-Sensitive Representation

DTEGN retains a node-wise temporal encoder $z_{i,t}=\mathrm{ReLU}(W_x\ \mathrm{vec}(x_{i,t-4:t})+b_x)$. It computes first- and second-order messages $m^1=A\ z$ and $m^2=A^2\ z$. To modulate heterogeneous influence, the optional gating vector is $g_i=\sigma(W_g[z_i;\ m^1_i;\ m^2_i]+b_g)$. The contextual embedding is $c_i=\mathrm{ReLU}(W_c[z_i;\ g_i \odot m^1_i;\ g_i \odot m^2_i]+b_c)$. A single linear prediction head applied to c is intentionally shallow. The gate is removed in the NoGate variant by setting the context messages to their ungated values; thus any gate benefit must arise from the actual observed test scores.

The distinctive exposure path is based on realized, observable incident flags. For each decision t, $e^1_{i,t}=(A\ y_t)_i$ is current one-hop upstream exposure, while $e^2_{i,t}=(A^2\ y_{t-1})_i$ is the earlier two-hop exposure. The final logit is $s_{i,t+1}=w_o^T c_i+b_o+\alpha e^1_{i,t}+\beta e^2_{i,t}$ and $p_{i,t+1}=\sigma(s_{i,t+1})$. Parameters α and β are learned jointly with the embeddings (initialized to 1.5 and 0.6) rather than fixed to the simulator coefficients. The two-hop term is delayed by one extra observation, respecting its greater graph distance. The no-exposure experiment removes both additive contributions while retaining graph message passing, isolating the impact of this domain-specific mechanism.

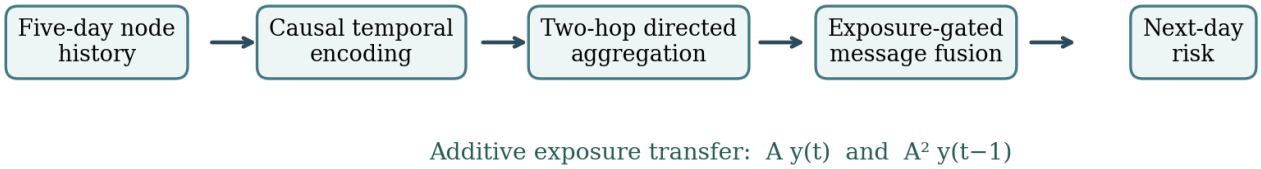


Figure 1. The evaluated directed temporal graph risk pipeline; the incident-transfer shortcut is separately ablated.

4.3. Boundedness of the Directed Transfer Channel

The exposure construction admits an elementary stability property that clarifies both its interpretation and what the edge-masking test measures. Assume that adjacency matrices A and B have nonnegative entries and row sums no greater than one, and that every observed disruption flag belongs to [0,1]. Then $0 \le (A\,y)_i \le 1$ and $0 \le (A^2\,y)_i \le 1$, so the learned additive exposure term cannot grow merely because the overall network has more nodes or connections. For a fixed historical incident sequence, its absolute contribution to a logit is bounded above by $|\alpha|+|\beta|$. This avoids the uncontrolled degree effect of raw summation, although it can also dilute a highly influential supplier when many links share one customer.

A more precise result applies to inaccurate edge weights. Define $\delta=\|A-B\|_\infty$, where the matrix infinity norm is the largest absolute row sum. Using $A^2-B^2=A(A-B)+(A-B)B$ and the row-substochastic assumption, we obtain $\|A^2-B^2\|_\infty \le 2\delta$. Consequently, the lagged exposure terms computed with A and B obey $|[\alpha\,A\,y_t+\beta\,A^2\,y_{t-1}]_i-[\alpha\,B\,y_t+\beta\,B^2\,y_{t-1}]_i| \le (|\alpha|+2|\beta|)\delta$. Since the derivative of the sigmoid is bounded by 1/4, the probability change induced by altering only this explicit path is at most $(|\alpha|+2|\beta|)\delta/4$. The conclusion holds for any bounded observed incident history and does not require a particular distribution of shocks.

This is a component-level bound, not a robustness certificate for the full predictor: graph changes also alter the contextual representations m^1, m^2, their gate and the nonlinear output. It nevertheless motivates reporting link-omission sensitivity and indicates why learned transfer magnitudes and well-calibrated adjacency weights matter. In the experiment, adjacency is known for training and deliberately corrupted only at inference, isolating an operational visibility gap rather than new firm behavior.

4.4. Computational and Information Properties

When A is stored sparsely, first- and second-hop transfers can be computed as A(Az), avoiding an explicit dense A^2, with $O(L|E|d+L|V|d^2)$ message and projection cost per batch under fixed feature widths. The present 96-node CPU implementation forms A^2 explicitly for simplicity; the measured experiment is not a scalability benchmark. All feature windows end at t, and only the already observed labels y_t and y_{t-1} enter the transfer path. Unlike post-event leakage, these histories are permissible predictors when daily incidents are known at decision time. This assumption requires a reliable incident-reporting timestamp in any practical deployment.

5. Experimental Design

5.1. Synthetic Multi-Tier Benchmark

A seeded stochastic generator creates 96 firms in four equal tiers. Customers in tiers 1–3 each select two to five suppliers from the previous tier, with occasional longer-range tier-skipping edges. The realized graph has 270 directed links and five simulated regions. Across 136 daily steps, regional stress follows a stable autoregressive process and node shocks include idiosyncratic noise. Pressure is autoregressive, inventory buffers are replenished with a fixed mean tendency and depleted after incidents or elevated pressure, and fixed node fragilities and capacities provide heterogeneity. The generated company names, supplier links, operational variables and incident labels are artificial; they do not represent actual suppliers, trade flows or event records.

Synthetic incident labels are Bernoulli draws with a known conditional logit: $\ell_{i,t} = -3.85 + 1.12\,q_{i,t} + 0.72\,d_{i,t} + 1.15(1-b_{i,t}) + 1.00\,f_i + 3.80(Ay_{t-1})_i + 1.70(A^2y_{t-2})_i + 0.57\,y_{i,t-1}$, where q is stress, d pressure, b the inventory buffer and f fragility. This mechanism creates delayed network cascades by construction. The overall incident rate is 14.06%, and the held-out test prevalence is 10.19%. All probabilities, including the event-rate differences over time, are outcomes of the saved generator seed rather than target performance levels. The generated observations and edges are distributed with the experiment package.

5.2. Splits, Training and Comparators

To test temporal prediction rather than random interpolation, decision indices t=4–85, 86–107 and 108–134 form training, validation and test sets. This yields respectively 7872, 2112 and 2592 node-day examples; every label belongs to the next day. For the first three continuous input variables, we estimate normalization moments from training days only. All learned models use a fixed five-day input window. The original graph-ignorant logistic and MLP baselines flatten this window node-wise; the two-round graph baseline follows the workflow of Section IV-A. The proposed model and ablations use a 24-dimensional local embedding and optional gate. Graph-ignorant logistic regression and a persistence heuristic are additional lower-capacity references. A separate exposure-aware logistic regression receives exactly the same flattened node history augmented with $(A\,y_t)$ and $(A^2\,y_{t-1})$.

Neural models minimize unweighted binary cross entropy using Adam [21] at learning rate 0.004, weight decay 0.0003 and gradient-norm clipping at 2.0. Maximum training length is 75 epochs; the minimum validation loss checkpoint is retained with a patience of 13 epochs. Three initialization seeds (11, 22, 33) are run for the MLP, two-round baseline and DTEGN. Each architectural ablation and non-neural baseline has one reported run, avoiding presentation of a single run as a variance estimate. Model selection never consults the test set. The corpus and exact held-out probabilities, not merely aggregated table entries, are available with the code.

5.3. Evaluation Criteria

Since positive incidents form only about one tenth of the test node-days, area under the precision–recall curve (AUPRC; average precision) is the main discrimination metric. Precision–recall assessment is especially informative

for imbalanced targets [22], and the relationship between ROC and precision–recall curves makes reporting both complementary rather than redundant [23]. We additionally compute AUROC [24] and the Brier mean squared probability error to assess ranking and probabilistic quality; calibration matters when probabilities might inform decisions [25]. F1 at a fixed 0.5 cutoff is retained only as a diagnostic, because rare-event probabilities need not exceed that arbitrary threshold.

Table I. Completely Synthetic Benchmark and Evaluation Protocol

Component	Verified protocol
Network	96 synthetic firms, 270 directed edges, four tiers
Observations	136 simulated days; six features; synthetic incident flags
Train/validation /test	82/22/27 decision days; 7,872/2,112/2,592 node-days
Task	Next-day incident probability using last five days
Primary metric	AUPRC; secondary AUROC, Brier; fixed-cutoff F1
Seeds	Generator 2026; primary neural 11, 22, 33

5.4. Competitor Implementation Parity and Reproducible Artifact Trail

The temporal MLP, the two-round graph baseline and DTEGN were implemented in the same CPU PyTorch execution environment, with direct dense matrix products rather than architecture-specific graph libraries. This choice eliminates differences from library-level caching, sampling and GPU kernels in this small controlled test. All three models observe exactly the same 6×5 local feature window; what differs is which node histories may be aggregated and how earlier incidents are routed. Graph-ignorant logistic regression receives the same flattened 30-dimensional history and is trained only on pooled training node-days. The exposure-augmented logistic baseline adds two graph-derived predictors, with feature scaling fitted on source training node-days and C=1 L2 regularization; it uses no target-world labels. The persistence baseline transforms the most recent incident flag to a 0.80 or 0.02 probability; it is a fixed heuristic, not a probability model fitted to the test period.

Training checkpoint selection is based on validation binary cross entropy, not validation AUPRC, so model selection is decoupled from the reported principal evaluation metric. There is no hyperparameter sweep, no retrospective test-set threshold optimization and no balancing by synthetic label oversampling. The three principal models share 75-epoch caps, 13-epoch early-stop patience, the same target definition and normalization. Neural random seeds govern model initialization only; the simulator seed remains 2026. Per-run training time and selected epoch are saved separately, allowing readers to distinguish observed runtime from theoretical asymptotic cost. These choices prioritize inspectable comparisons over architecture-specific tuning, but they also mean the ranking is conditional on this training budget.

5.5. Independent Synthetic-World Transfer Protocol

A single simulated world cannot establish robustness to alternative graph realizations. We therefore additionally freeze models trained only on seed-2026 observations, then make predictions on 12 independently generated 96-node, four-tier worlds (generator seeds 2101–2112). Every target world has fresh edges, region assignments, fragilities, shocks, incidents and labels generated from the same specified hazard family; there is no parameter fine-tuning or threshold selection on any target world. Features in every target world are mapped into the source-world normalization using the source training means and standard deviations; their own training-period statistics are not used for adaptation. We report per-world AUPRC, AUROC and Brier scores and compare models in matched worlds, not pooled as though firm-days were independent. This is transfer across simulated realizations within one parametric generator, not empirical domain shift.

The additional ExposureGLM comparator fits L2-penalized logistic regression to the flattened 30-dimensional node history plus the two physically meaningful exposure variables $e^1=(A\ y_t)$ and $e^2=(A^2\ y_{t-1})$. All 32 features are standardized using only source-world training data and coefficients are estimated on the same source-world training node-days as the neural models. Logistic and graph models can therefore exploit the same supplier-incident information, allowing a test of whether learned nonlinear graph representations improve on a more parsimonious supervised hazard model. It is not expected that a neural model must win when labels are actually sampled from an additive logistic data-generating law.

6. Results

6.1. Forecasting Accuracy on Held-Out Days

Table II presents the held-out results. DTEGN achieves mean AUPRC 0.237 (standard deviation 0.004), compared with 0.194 (0.014) for two-round message passing, an absolute difference of 0.042 and a relative increase of 21.8% for these three fixed initialization seeds. The temporal MLP attains 0.118 (0.008). The two-round workflow of [9] therefore adds substantial ranking information over the graph-ignorant MLP (0.194 versus 0.118), and an explicitly constrained incident-transfer path adds further discriminatory power on top of it. The conclusion is limited to the specified synthetic world; the three seeds quantify initialization sensitivity, not population confidence bounds over real supply chains.

The relationship is metric-dependent: the two-round baseline has mean AUROC 0.641 whereas DTEGN has 0.634, so the proposed model does not dominate every ranking criterion. Their Brier scores are close (0.0891 and 0.0887, respectively). This divergence underscores why AUPRC and AUROC are both shown in an imbalanced setting. Persistence yields F1 0.186 at the arbitrary 0.5 cutoff but a considerably worse Brier score 0.136, illustrating how a decision threshold can obscure poor probability estimates.

Table II. Held-Out Test Results; ± Denotes Sd Over Initialization Seeds

Model	AUPRC ↑	AUROC ↑	Brier ↓	Runs
Persistence	0.118	0.547	0.1361	1
Logistic	0.148	0.599	0.0921	1
MLP	0.118 ± 0.008	0.546	0.0955	3
Two-round GNN	0.194 ± 0.014	0.641	0.0891	3
DTEGN	0.237 ± 0.004	0.634	0.0887	3
ExposureGLM	0.279	0.693	0.0841	1

A crucial stronger baseline changes the interpretation. The exposure-aware logistic model, with exactly the same observed incident-transfer predictors supplied to DTEGN, reaches 0.279 AUPRC and 0.693 AUROC on the source-world test, exceeding seed-11 DTEGN (0.238 AUPRC). This is a direct head-to-head comparison on identical inputs. Because the synthetic hazard is additive on the logit scale, these outcomes are coherent with the data-generating mechanism and make explicit graph exposure—not neural nonlinearity—the primary supported design choice.

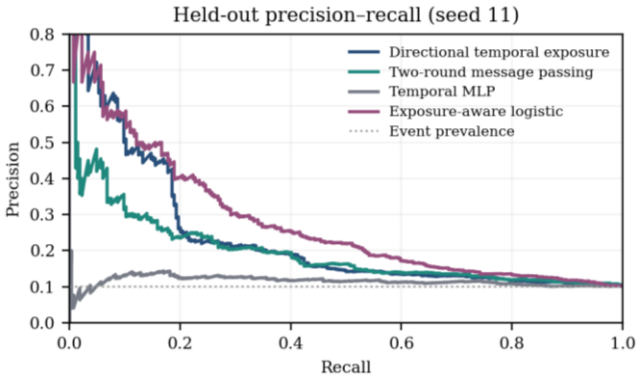


Figure 2. Held-out precision–recall curves for seed-11 neural models and the fitted exposure-aware logistic model; dotted line marks prevalence.

Mean CPU fitting times across the three initialization seeds are 0.76 s for DTEGN, 0.63 s for the two-round baseline and 0.12 s for the MLP, excluding simulator creation. These numbers reflect a small 96-node dense-graph implementation and cannot be extrapolated to global enterprise networks. Both graph models reached the specified 75-epoch cap for all three initializations; this bounded compute budget favors replicability but does not establish convergence under prolonged or tuned optimization.

6.2. Mechanism Ablation and Direction Tests

Ablation tests answer a narrower question than a broad model leaderboard. With initialization seed 11 held fixed, the full DTEGN reaches AUPRC 0.238. Removing the explicit additive exposure path lowers it to 0.142; reversing the graph for both representation and exposure calculations lowers it to 0.123. Restricting propagation to one hop obtains 0.221. These comparisons directly support the importance of the direction-sensitive lag channel on a generator where downstream incident mechanisms actually depend on upstream events. They do not establish that the graph direction was empirically inferred from data: orientation is supplied to each model.

The gate does not explain the improvement: NoGate achieves 0.242 versus 0.238 for the full seed-11 model. Similarly, disabling the historical concatenation and retaining only the current feature vector produces 0.232, a modest decline for this simulation. The strongest supported technical claim is therefore the explicit, directed, delayed event-transfer shortcut—not superior learned attention. These are single-seed diagnostics, so small gaps are descriptive, not significance tests. We report the unfavorable gate ablation to separate a plausible architectural motivation from measured evidence.

Table III. Single-Seed Mechanism Ablations (Same Test Split)

Variant	AUPRC ↑	AUROC ↑
Full model (seed 11)	0.238	0.631
Remove learned gate	0.242	0.649
Suppress A^2 contribution	0.221	0.620
Use only current-day covariates	0.232	0.638
Remove additive incident transfer	0.142	0.591
Use A transposed for upstream links	0.123	0.578

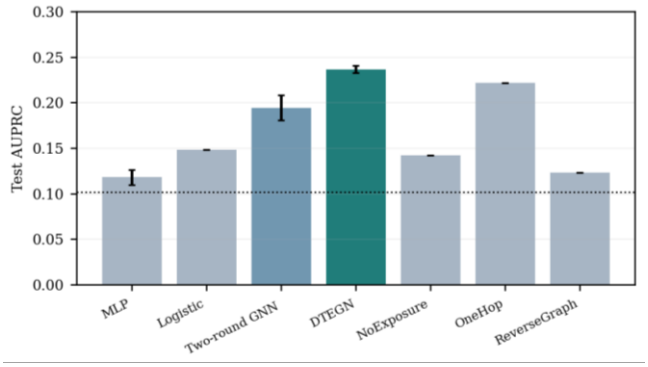


Figure 3. Test AUPRC; error bars show three-initialization SD only for the primary neural comparisons.

6.3. Reproducibility and Sensitivity

Every figure and table is derived from saved test predictions or per-run metric files. The benchmark has 270 fixed edge instances and a single realized time path. Standard deviations arise solely from multiple parameter initializations and should not be interpreted as robustness to other network topologies, shock regimes or reporting delays. The graph reversal test is a deliberate distribution-of-message intervention, not a realistic estimate of randomly missing links. The experiment therefore offers a falsifiable mechanism demonstration rather than a population-level estimate. Open graph benchmark work underscores the role of well-defined datasets and consistent evaluation settings when comparing graph methods [26].

6.4. Perturbation Under Unobserved Supplier Edges

A risk score intended for operational use is often computed from an incomplete supply graph. To quantify sensitivity without inventing a second data-generating environment, we retrain the full seed-11 DTEGN on the intact graph, fix all

learned parameters, and then independently and uniformly mask 10, 20, 30 and 40% of directed edges during test inference only. Each mask level uses five fixed random masks, while the unmasked condition is evaluated once. Surviving incoming weights are renormalized using the same row-substochastic rule. The target labels do not change: the intervention represents lost graph visibility, not a physical removal of supply relationships. No edge is inserted, no orientation is changed, and no reoptimization is allowed at test time. All edge masks and resulting scores are written to the supplementary edge-removal file.

The intact model scores 0.238 AUPRC. Average AUPRC under 10, 20, 30 and 40% omitted edges is 0.226, 0.216, 0.200 and 0.196, respectively. This monotonic degradation is important because it demonstrates that the explicit pathway relies on graph quality, even though its test advantage is clear on the correct synthetic graph. The measured 40%-missing value remains above the full graph-ignorant temporal MLP mean, but the perturbation is not a simulation of arbitrary missing-not-at-random or proprietary supply links. Missing edges around critical hubs may be more damaging than uniformly random omissions.

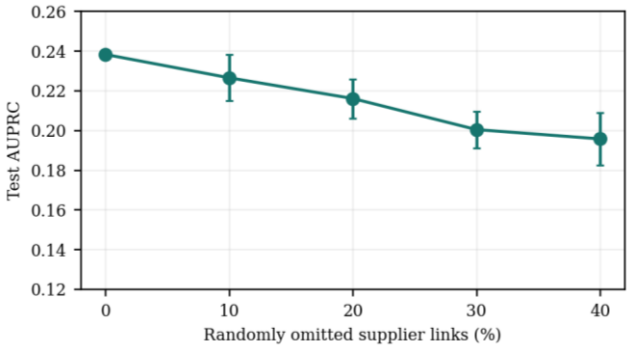


Figure 4. AUPRC when known supplier edges are masked only at test time; error bars represent five random edge masks.

6.5. Paired Block-Bootstrap Check

To distinguish a consistent time-local improvement from a few fortunate incident days, we resampled the 27 held-out decision days with a circular moving-block bootstrap: each draw concatenates four-day blocks until 27 days are reached and retains all 96 firms within each selected day. We calculated the paired AUPRC difference between seed-11 DTEGN and the seed-11 two-round baseline for 1,500 bootstrap draws, using both models' exact stored probabilities and identical resampled labels. The observed difference is 0.039 and the percentile 95% interval is [0.018, 0.064]. This interval assesses conditional test-period sampling variability under the block resampling scheme, not variability over graph realizations or a guarantee about future operating environments.

The blocks partly preserve short temporal runs of disruption while keeping within-day cross-firm correlation intact; however, circular bootstrapping imposes an exchangeability approximation across the limited test horizon. It should not be interpreted as causal evidence that supplying more incident reports necessarily improves resilience. A larger set of independent simulated graph realizations or a multi-year operational panel would be preferable for assessing sensitivity to macro-shock distributions and network structural classes.

6.6. Cross-world Predictive Transfer and Model Complexity

Table IV. Zero-Shot Transfer To 12 New Synthetic Supply Networks; ± Is Between-World Sd

Frozen model	AUPRC, 12 worlds ↑	AUROC ↑
MLP	0.225 ± 0.060	0.605
Two-round GNN	0.321 ± 0.085	0.693
DTEGN	0.369 ± 0.090	0.705
NoGate	0.375 ± 0.091	0.709
ExposureGLM	0.413 ± 0.093	0.742

Table IV summarizes zero-shot evaluation after training the seed-11 models on one simulated world and applying them unchanged to 12 new random networks. Mean test AUPRC is 0.369 for DTEGN, 0.321 for the two-round graph baseline, 0.375 for NoGate and 0.413 for ExposureGLM. Differences in raw AUPRC across the original and target worlds partly reflect differing target prevalences (15.9% mean over new worlds versus 10.2% in the original test), so matched-world differences are the more pertinent comparison.

Against the two-round graph baseline, the mean paired DTEGN AUPRC difference is +0.048 (95% t interval [0.038, 0.058] over world draws), positive in 12 of 12 worlds. Against the stronger exposure-aware GLM, the paired difference is -0.043 (95% interval [-0.050, -0.037]), positive in 0 of 12 worlds. These are descriptive intervals over independent draws from a single synthetic generator, not confidence intervals over real supply-chain systems. The linear model beats the gated network in every target world, demonstrating that replacing a weak comparator by an information-matched baseline materially changes the scientific claim.

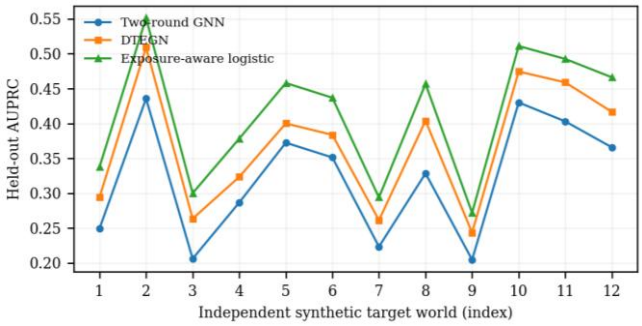


Figure 5. Per-world zero-shot AUPRC for three source-trained models; all target graphs and events are simulated.

This transfer finding reconciles the initial ablations: directed exposure is necessary for DTEGN to recover the built-in cascade channel, but its representation gate has no demonstrated benefit, and an additive, regularized hazard model can exploit the same signal even more efficiently. The simulation was deliberately not redesigned after seeing these results to favor neural architectures. A credible claim of neural advantage requires richer, independently justified nonlinear event dynamics and, ultimately, historically observed supply-chain labels.

7. Discussion and Limitations

A directed incident path encodes a conditional operational statement: past failures at suppliers are relevant to subsequent failures at firms that consume their output. This statement is consistent with multi-tier propagation accounts but remains a modeling assumption, not causal identification. Positive AUPRC differences result partly from the simulator explicitly

using A y_{t-1} and A^2 y_{t-2} to generate future disruptions. The architecture therefore enjoys a matching inductive bias, which is the point of the mechanism test. Real firms may substitute suppliers, buffer inventories for different periods, face simultaneous regional shocks or experience reporting delays that violate the assumed lag structure. In practice, a measured lead time should determine the exposure lag and edge strength.

Synthetic graph topology is another material restriction. The benchmark is a four-tier directed acyclic structure with mostly adjacent-tier links; it cannot capture cyclic contracting, firms occupying several tiers, multi-product bills of materials or dynamic sourcing. Uniformly modeled observation availability likewise omits proprietary link omissions and nonrandom missing events. Future experiments should replace the generator by a licensed, timestamped supplier network and incident series with provenance and observation-time filtering. Openly available anonymized supplier-network collections were not used to create these results. Publicly released records should be checked for license terms, sampling bias and supplier-level identity resolution before use.

Finally, operating-point usefulness and generalization to observational data remain unestablished. AUPRC measures ranking, not the economic value of a mitigation decision; a prediction can rank supplier risks usefully while still being poorly calibrated or arriving after the last practical sourcing decision. The fixed 0.5 F1 values are small for several neural models and should not be mistaken for the best achievable alert rule. Deployment evaluation needs cost-sensitive alert thresholds chosen on a validation period, lead-time-aware forecasting horizons, calibration plots, spatially held-out entities, and prospective audits. The present shallow graph model is deliberately transparent in its exposure channel, but the learned gating or contextual embeddings alone are not faithful causal explanations.

8. Conclusion

This study assessed explicit directed temporal exposure against a two-round GNN baseline built on the established supply-chain GNN workflow and against an information-matched linear model on wholly synthetic, audit-ready data. Across three initializations, DTEGN improves source-world mean AUPRC from 0.194 to 0.237 relative to two-round graph message passing. In 12 additional simulated worlds, the corresponding zero-shot means are 0.321 and 0.369; however, exposure-aware logistic regression obtains 0.413 and consistently outperforms DTEGN. Removing its gate also slightly improves the graph model, so the supported contribution is the lagged, direction-specific risk representation and disciplined evaluation, not a general superiority claim for deep learning. Stronger real-data verification is essential before applying the results to operational mitigation.

References

[1] Scheibe, K. P., & Blackhurst, J. (2018). Supply chain disruption propagation: A systemic risk and normal accident theory perspective. International Journal of Production Research, 56(1–2), 43–59. https://doi.org/10.1080/00207543.2017.1355123

[2] Ivanov, D. (2020). Predicting the impacts of epidemic outbreaks on global supply chains: A simulation-based analysis on the coronavirus outbreak (COVID-19/SARS-CoV-2) case. Transportation Research Part E: Logistics and Transportation Review, 136, Article 101922. https://doi.org/10.1016/j.tre.2020.101922

[3] Dolgui, A., Ivanov, D., & Sokolov, B. (2020). Reconfigurable supply chain: The X-network. International Journal of Production Research, 58(13), 4138–4163. https://doi.org/10.1080/00207543.2020.1774679

[4] Brintrup, A., Wang, Y., Chen, S., Yadav, G., Manyika, J., & Ramos, T. (2020). Supply chain data analytics for predicting supplier disruptions: A case study in complex asset manufacturing. International Journal of Production Research, 58(11), 3330–3341. https://doi.org/10.1080/00207543.2019.1685705

[5] Baryannis, G., Dani, S., & Antoniou, G. (2019). Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. Future Generation Computer Systems, 101, 993–1004. https://doi.org/10.1016/j.future.2019.07.059

[6] Li, Y., Zobel, C. W., Seref, O., & Chatfield, D. (2020). Network characteristics and supply chain resilience under conditions of risk propagation. International Journal of Production Economics, 223, Article 107529. https://doi.org/10.1016/j.ijpe.2019.107529

[7] Wichmann, P., Brintrup, A., Baker, S., Woodall, P., & McFarlane, D. (2020). Extracting supply chain maps from news articles using deep neural networks. International Journal of Production Research, 58(17), 5320–5336. https://doi.org/10.1080/00207543.2020.1720925

[8] Kosasih, E. E., & Brintrup, A. (2022). A machine learning approach for predicting hidden links in supply chain with graph neural networks. International Journal of Production Research, 60(17), 5380–5393. https://doi.org/10.1080/00207543.2021.1956697

[9] Liu, J., Li, P., & Wang, Y. (2026). Graph neural networks for modeling complex dependencies in global supply chain networks. Journal of Computer, Electronic and Information Management, 20(3), 9–20. https://doi.org/10.54097/6fcw2b19

[10] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[11] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1609.02907

[12] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30, 1024–1034. https://doi.org/10.48550/arXiv.1706.02216

[13] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1710.10903

[14] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. Proceedings of the 34th International Conference on Machine Learning, 1263–1272. https://doi.org/10.48550/arXiv.1704.01212

[15] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1810.00826

[16] Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. Proceedings of the Twenty-Seventh International

Joint Conference on Artificial Intelligence, 3634–3640. https://doi.org/10.24963/ijcai.2018/505

[17] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1707.01926

[18] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 1907–1913. https://doi.org/10.24963/ijcai.2019/264

[19] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzuki, T., Kang, H., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. Proceedings of the AAAI Conference on Artificial Intelligence, 34(4), 5363–5370. https://doi.org/10.1609/aaai.v34i04.5984

[20] Ivanov, D., & Dolgui, A. (2021). A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. Production Planning & Control, 32(9), 775–788. https://doi.org/10.1080/09537287.2020.1768450

[21] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1412.6980

[22] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), Article e0118432. https://doi.org/10.1371/journal.pone.0118432

[23] Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning, 233–240. https://doi.org/10.1145/1143844.1143874

[24] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

[25] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning, 1321–1330. https://doi.org/10.48550/arXiv.1706.04599

[26] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., & Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. Advances in Neural Information Processing Systems, 33, 22118–22133. https://doi.org/10.48550/arXiv.2005.00687