The Network Foundations of Credit Counterparty Risk: Theory and Evidence *

Jonathan Brogaard† Belinda (Chen) Chen ‡

Abstract

We study how supply-chain relationships shape corporate credit risk. We develop a CES production-network model in which sectoral productivity shocks propagate through direct and high-order input-output linkages and affect firms' distance to default. Credit spreads depend on the full network generated by primitive trade weights, substitution elasticities, and equilibrium prices. Finite shocks to central suppliers can trigger downstream defaults in hub-and-spoke bottlenecks, while extreme supplier defaults become economy-wide credit-risk events when inputs are sufficiently complementary. Empirically, we construct supply-chain-disciplined idiosyncratic-risk spillover networks and show with graph machine learning tools that network topology improves CDS-spread prediction relative to a zero-edge benchmark, especially for network-sensitive sectors.

Keywords: Credit Counterparty Risk, Default Risk, Production Network, Idiosyncratic Risk Spillover

JEL Classification: C45, C58, G12, G17

*This paper was previously circulated under the title 'Attention-Based Graph Neural Networks in Firm CDS Prediction'. We are grateful for insightful comments and suggestions from Jack Bao (discussant), Darrell Duffie, Tim Johnson, Dana Kiku, Kayla Freeman (discussant), Neil Pearson, Jun Pan, Guillaume Roussellet (discussant), and Rui Zhong (discussant). We have also benefited greatly from feedback received at the CICF, AFA and WFA conferences.

†David Eccles School of Business, University of Utah;brogaardj@eccles.utah.edu

‡Shanghai Advanced Institute of Finance (SAIF), Shanghai Jiao Tong University; chenchen@saif.sjtu.edu.cn

1 Introduction

The interdependence of firms' default risk is an important topic in the study of credit risk. A large body of literature shows that default clustering arises mainly for two reasons. First, firms are exposed to common shocks or correlated risk factors, so default probabilities are jointly driven by aggregate macroeconomic and financial conditions (e.g., Duffie and Singleton (1999), Collin-Dufresne, Goldstein, and Martin (2001), and Duffie and Gârleanu (2001)). Second, firms may experience downside firm-specific shocks that propagate to other firms through economic linkages, generating contagion within firm networks. This type of idiosyncratic risk spillover is also referred to as counterparty risk.

Existing theoretical studies of counterparty risk have provided important insights into how default spillovers arise, but two aspects remain less fully developed. First, firms are linked through multiple economic channels, yet many models do not explicitly specify the underlying linkage mechanism. Second, many models are tractable only in highly simplified settings, such as two-firm economies, and therefore typically abstract from cascading and feedback effects. We build on this strand of literature by providing a comprehensive framework for understanding how the inter-firm network shapes risk spillovers.

In this paper, we do two main things. First, we propose a general credit-risk framework that microfounds counterparty risk in production-based networks. The framework accommodates a large number of firms connected through input–output relationships, allows for cascading effects and asymmetric contagion, and explicitly incorporates the full network topology. We use the framework to derive new implications for how the entire network affects firms' distance to default and credit spreads. Second, we develop a parsimonious empirical approach based on graph machine learning. The graph neural network is disciplined by the model mechanism and provides direct empirical support for the model implications.

We develop a structural model in which each firm is also a production sector. Firm i produces good i and uses goods produced by other firms as intermediate inputs. Throughout the structural model, the index i denotes both firm i and the production sector associated with good i. This notation keeps the production block and the credit-risk block at the same level of aggregation. A negative productivity shock to supplier firm j raises the effective cost of using input j. This effective cost can be interpreted as a higher product price, a higher scarcity shadow price, or lower input availability. Customer firms that use input j then face weaker fundamentals: higher input costs, lower operating cash flows, lower equity values, higher equity volatility, and lower distance to default.

The model begins with a CES production economy. The primitive input-output matrix $W_t = [\omega_{ij,t}]$ records direct input weights. Element $\omega_{ij,t}$ is the proportion of customer firm i's direct intermediate-input bundle assigned to supplier good j. We impose row normalization on these primitive input weights. The substitution elasticity σ_i is a primitive parameter of firm i's production technology. We assume that intermediate inputs are gross complements when $\sigma_i < 1$, gross substitutes when $\sigma_i > 1$, and Cobb–Douglas in the limit as $\sigma_i \to 1$. This elasticity determines whether a customer firm can substitute away from a disrupted input. A supplier price increase is amplified when the disrupted input is complementary to other inputs and attenuated when the customer can substitute toward alternative inputs.

The primitive input-output network generates two high-order objects. The first object is the level exposure matrix $\ell_t = [\ell_{ij,t}]$. It is obtained from a local CES propagation matrix built from equilibrium expenditure shares. The matrix ℓ_t is not the standard Leontief inverse used in production-based multisector models to solve for sectoral sales shares, Domar weights, or gross-output responses. The standard Leontief inverse is built from primitive input cost shares such as $\alpha_i \omega_{ij,t}$. By contrast, ℓ_t is built from the realized CES expenditure shares $s_{ij,t}$ and measures the first-order and higher-order pass-through of supplier shocks through those equilibrium shares. We use this multiplier to study how the production network affects customer revenues, cash flows, equity values, and cash-flow volatility. The element $\ell_{ij,t}$ is the total pass-through of a productivity shock originating at supplier firm j to customer firm i, including both the direct supplier-customer link and higher-order input-output paths.

The second object is the volatility exposure. Cash-flow volatility is a second moment, so supplier j's contribution to customer i's cash-flow variance is quadratic in the level exposure. Under conditional orthogonality of idiosyncratic supplier shocks, the raw variance contribution is $v_{ij,t} = \ell_{ij,t}^2 \sigma_{u,j,t}^2$, where $\sigma_{u,j,t}^2$ is supplier j's idiosyncratic shock variance. In the empirical implementation, this variance is proxied by supplier idiosyncratic volatility, $IVOL_{j,t}^2$. Thus, the model distinguishes a cash-flow level channel, governed by $\ell_{ij,t}$, from a cash-flow volatility channel, governed by $\ell_{ij,t}^2 IVOL_{j,t}^2$.

We then map these network-driven fundamentals into default risk. Firm i has asset value, asset volatility, and zero-coupon debt with face value F_i maturing at T. Under the Merton distance-to-default framework, default occurs at maturity when asset value falls below the debt face value. Because asset value and asset volatility are not directly observed, we follow Bharath and Shumway (2008) and approximate them using equity value, debt face value, and equity volatility. The production network affects equity value through the cash-flow level channel and affects equity volatility through the cash-flow volatility channel. Combining

these mappings yields a risk-neutral distance to default that is a nonlinear function of the full input-output network. Given a fixed recovery rate, distance to default maps into risk-neutral default probability and CDS spreads.

The model yields three implications. First, the full network structure prices the cross section of CDS spreads nonlinearly. Orthogonal negative idiosyncratic productivity shocks propagate through both direct and indirect input-output relationships, affect equity value and equity volatility, and enter distance to default through both the level and volatility channels. Second, detailed network topology determines the severity of finite-shock spillovers. In a hub-and-spoke structure where a dominant supplier provides a large share of inputs to many customer firms and also relies heavily on its own output, a finite shock to the supplier can push customer firms close to, or into, actual default even if the supplier itself remains solvent. The same topology can feed customer distress back to the supplier through weak input links from peripheral firms to the hub. Third, when a supplier experiences an infinitely large productivity shock that directly drives it to default, the elasticity of substitution determines whether that actual default becomes an economy-wide credit-risk event. Complementary input links transmit the disruption because the affected input becomes a bottleneck; substitutable input links can block propagation because customers can reallocate away from the disrupted input.

Empirically, we use graph neural networks (GNNs) to implement the model-implied network exposures in CDS-spread prediction. Direct structural estimation of a large production-network model is computationally difficult, especially when the model contains an $N \times N$ network matrix, nonlinear propagation, and potentially high-order cascade effects. GNNs provide a natural empirical tool for this setting because they are designed to learn from data observed on graphs. Specifically, GNNs take the full network topology as input, including firm-level node characteristics and inter-firm edge characteristics, and use this information to predict firm-level CDS spreads.

The GNN architecture mirrors the economic mechanism in the model. In the message-passing step, each firm's representation is updated using information from connected firms, with the contribution of each neighbor weighted by the corresponding edge characteristic. This captures the idea that a firm's credit risk depends not only on its own fundamentals, but also on the strength of its supply-chain exposures. Stacking multiple GNN layers allows information to propagate through direct and indirect network paths, mimicking the first-order and high-order terms in the network multiplier that governs risk transmission in the model.

We train the GNN to predict firm-level CDS spreads. We also train a benchmark model in which all

network edges are removed while preserving the same node characteristics and training protocol. This zero-edge benchmark provides a clean comparison for measuring the incremental value of network structure. In the empirical implementation, the model-implied firm-level network is measured using industry-level supply-chain spillovers, which are then imposed on firms according to their industry classifications. Thus, the GNN does not estimate the structural model directly, but provides an empirical implementation of its central implication: supply-chain network topology contains economically meaningful information for pricing corporate credit risk.

The empirical results indicate that network topology is quantitatively important for credit spreads. Under identical training and validation protocols, the GNN substantially reduces out-of-sample prediction error relative to the zero-edge benchmark. Economically, incorporating network edge features induces an average network-attributable spread change of approximately 21.8% and yields an incremental R^2 of 0.56 relative to the zero-edge benchmark. Network features become especially valuable during the 2008 financial crisis, the post-crisis network repair period, the 2018 trade-tension episode, and the COVID-19 supply-chain disruption. Industry heterogeneity in network-attributable spread changes further supports the model's substitution-elasticity mechanism. Network information is most important for sectors such as electrical equipment, shipbuilding and railroad equipment, and defense-related industries, where production often depends on specialized, certified, or system-specific inputs that are difficult to replace quickly. By contrast, network information changes predicted spreads less for sectors such as textiles, rubber and plastics, mining, coal, and entertainment, where supplier inputs are more substitutable or credit risk is more strongly driven by final demand or firm-specific fundamentals.

We contribute to the literature primarily by providing a structural and conceptual framework that clarifies how production-network linkages shape firms' credit spreads. On the theory side, we provide, to the best of our knowledge, the first comprehensive analysis of how the full production-network structure affects corporate credit risk through both direct and high-order spillover channels. Existing theoretical work on credit counterparty risk has made important progress in showing how idiosyncratic risk spillovers can generate excess default clustering beyond what can be explained by common factors. We build on this literature by specifying a concrete economic mechanism: input-output relationships. This allows us to characterize how supply-chain linkages transmit shocks across firms and how the full network multiplier enters credit-spread pricing.

Our framework also complements existing structural models by moving beyond highly stylized low-

dimensional spillover settings. Many structural models adopt simplified two-firm or small-network structures without cascading effects for tractability. These frameworks provide valuable intuition, but they make it difficult to analyze the role of large-scale interconnectedness and high-order propagation. In contrast, our framework studies how the entire network structure shapes credit risk. This distinction is important because features such as supplier concentration, centrality, and high-order network paths can materially affect the severity of risk transmission.

On the empirical side, we provide an empirical framework that incorporates full network topology into CDS-spread prediction using graph neural networks (GNNs). This approach allows us to incorporate firm-level node characteristics and inter-firm edge characteristics in a unified predictive model. We also recognize a growing and promising literature that applies graph-based machine-learning tools to asset pricing and financial networks, such as Capponi, Sidaoui, and Zou (2025). Our contribution is to apply these tools under the discipline of a structural production-network framework. In particular, the GNN architecture is not used only as a flexible prediction device; rather, its message-passing mechanism closely mirrors the economic mechanism in the model, where shocks propagate through direct and high-order network linkages. This connection allows us to use GNNs as an empirical implementation of the model-implied network channel in credit-spread pricing.

Related Literature

This paper contributes to the literature on credit risk and default contagion. A large body of work studies default clustering, credit-risk spillovers, and interconnectedness in credit markets. One strand emphasizes common risk factors and aggregate state variables (e.g., Collin-Dufresne, Goldstein, and Martin (2001) and Duffie and Gârleanu (2001)). Another strand studies idiosyncratic contagion and counterparty risk, including Jarrow and Yu (2001), Das, Duffie, Kapadia, and Saita (2007), Jorion and Zhang (2007), Jorion and Zhang (2009), and Azizpour, Giesecke, and Schwenkler (2018). We contribute to the second strand by providing a production-network mechanism for idiosyncratic default-risk propagation. The model allows ordinary idiosyncratic shocks, rather than only rare firm-specific tail events, to propagate through higher-order economic linkages. It also permits asymmetric contagion and cascading effects in a network with many firms.[1] While some CDS-focused studies, such as Getmansky, Girardi, and Lewis (2016), emphasize

[1] Related contributions include Hawkes (1971), Giesecke (2002), Giesecke and Weber (2004), Giesecke (2004), Kitwiwattanachai and Pearson (2015), Benzoni, Collin-Dufresne, Goldstein, and Helwege (2015), Aït-Sahalia, Cacho-Diaz, and Laeven (2015), and Jacobson and Von Schedvin (2015).

counterparty risk arising from common protection sellers or dealer networks, we abstract from dealer cores and OTC market structure and focus instead on real economic linkages through supply chains.

We also contribute to the broader literature on the measurement and pricing of credit risk. This literature includes structural and reduced-form models such as Merton (1974), Duffie and Singleton (1999), Duffie (1999), Duffie, Pan, and Singleton (2000), Duffie and Lando (2001), Duffie and Pan (1997), Duffie, Pedersen, and Singleton (2003), Duffie, Saita, and Wang (2007), Bharath and Shumway (2008), Bao and Pan (2013), and Hu, Pan, and Wang (2013). Our framework uses a distance-to-default mapping to connect real supply-chain shocks to default probabilities and CDS spreads.[2] Our focus differs from models that emphasize non-default credit premia, time-varying risk premia, or rare-disaster risk. We instead ask how the cross-sectional production-network structure governs the propagation of idiosyncratic default risk.

The paper is closely related to the production-network literature. A large body of work studies how input-output linkages affect aggregate productivity, aggregate volatility, and macroeconomic tail risk, including Carvalho (2008), Acemoglu, Carvalho, Ozdaglar, and Tahbaz-Salehi (2012), Carvalho and Gabaix (2013), Herskovic (2018), and Chen (2023). We adapt this production-based input-output framework to study credit spreads. The distinction between primitive input weights, high-order level exposures, and quadratic volatility contributions follows the logic that direct and indirect production paths jointly determine shock propagation. Our elasticity-based implication is also related to Dew-Becker (2023), who show that nonlinear production networks can have different local and tail propagation properties.[3]

Finally, this paper contributes to the growing literature on machine learning in finance. Studies such as Kelly, Pruitt, and Su (2019), Gu, Kelly, and Xiu (2020), and Gu, Kelly, and Xiu (2021) show that machine-learning methods are useful for high-dimensional prediction problems in asset pricing and corporate finance. We use graph neural networks as an empirical tool for incorporating granular network topology into credit-spread prediction.[4] The purpose of the GNN is not only to improve prediction, but also to implement the model-implied network exposure that maps supply-chain disruption into firm-level credit risk.

[2] Additional related studies include Altman (1968), Duffie and Liu (2001), Almeida and Philippon (2007), Campbell, Hilscher, and Szilagyi (2008), Longstaff, Pan, Pedersen, and Singleton (2011), Ang and Longstaff (2013), Gouriéroux, Monfort, and Renne (2014), Galil, Shapir, Amiram, and Ben-Zion (2014), Kitwiwattanachai (2015), Berndt, Douglas, Duffie, and Ferguson (2018), Boyarchenko and Shachar (2020), Monfort, Pegoraro, Renne, and Roussellet (2021), and Bao, Hou, and Zhang (2023).

[3] Additional related literature includes Gabaix (2011), Diebold and Yılmaz (2014), Acemoglu, Akcigit, and Kerr (2016), Blasques, Koopman, Lucas, and Schaumburg (2016), Härdle, Wang, and Yu (2016), Acemoglu, Ozdaglar, and Tahbaz-Salehi (2017), Demirer, Diebold, Liu, and Yilmaz (2018), Chen, Härdle, and Okhrin (2019), Liu (2022), Dew-Becker (2023), Engle and Kelly (2012), Herskovic, Kelly, Lustig, and Van Nieuwerburgh (2016), and Herskovic, Kelly, Lustig, and Van Nieuwerburgh (2020).

[4] Additional related work includes Kelly and Jiang (2014), Kelly, Malamud, and Zhou (2024), Wang, Lin, Cui, Jia, Wang, Fang, Yu, Zhou, Yang, and Qi (2019), Uddin, Tao, and Yu (2021), and Zhang, Pu, Cucuringu, and Dong (2023).

The remainder of the paper is organized as follows. Section 2 presents the structural framework. Section 3 describes the empirical design and results. Section 4 concludes.

2 Structural Framework

We develop a production-based model to microfound credit counterparty risk through input-output relationships. The model follows the sequence from network structure to firm fundamentals, from firm fundamentals to distance to default, and from distance to default to credit spreads.

There are N firms. Firm i produces good i and purchases goods produced by other firms as intermediate inputs. We use i to denote both the firm and the production sector associated with that firm's good. A negative productivity shock to one firm directly weakens that firm's production and can also transmit to customer firms that use its good as an input. The same shock can then transmit to customers of those customers through higher-order input-output paths.

The production network shapes fundamentals because operating cash flow depends on revenue net of input costs. First-order input-output links affect direct customers. Higher-order links affect firms connected through longer production paths. We characterize how the full input-output network maps firm-specific productivity shocks into cash-flow levels and cash-flow volatility.

We then map these fundamentals into default risk using a Merton distance-to-default framework. Firm i has asset value, asset volatility, and zero-coupon debt with fixed face value and maturity. Default occurs when the firm's asset value falls below the face value of debt at maturity. Following Bharath and Shumway (2008), we approximate asset value with equity value plus debt face value, and we approximate asset volatility using equity value, debt face value, and equity volatility. Because equity value is supported by operating cash flows, the production network affects distance to default through both a cash-flow level channel and a cash-flow volatility channel.

Combining these components gives firm i's risk-neutral distance to default as a nonlinear function of cash-flow value, cash-flow volatility, and the full input-output network. Given a fixed recovery rate, the model maps distance to default into default probability and CDS spreads. The framework therefore identifies how direct and higher-order production-network linkages affect credit spreads through fundamentals and default risk.

2.1 Production Network

There are N firms indexed by $i, j = 1, \ldots, N$. Firm i produces output $Y_{i,t}$ using labor $L_{i,t}$ and a CES aggregate of intermediate inputs. Let $X_{ij,t}$ denote the quantity of good j used by firm i as an intermediate input. The production function is

$$Y_{i,t} = \xi_t Z_{i,t} L_{i,t}^{1-\alpha_i} \left(\sum_{j=1}^{N} \omega_{ij,t}^{1/\sigma_i} X_{ij,t}^{(\sigma_i-1)/\sigma_i} \right)^{\alpha_i \sigma_i/(\sigma_i-1)}, \qquad 0 < \alpha_i < 1,$$

where ξ_t is aggregate productivity, $Z_{i,t} = \exp(z_{i,t})$ is firm-specific productivity, $z_{i,t}$ is log productivity, $\omega_{ij,t} \geq 0$ is the primitive input weight from supplier good j to customer firm i, α_i is firm i's intermediate-input share, and $\sigma_i > 0$ is firm i's elasticity of substitution across intermediate inputs. The parameter σ_i governs substitutability across intermediate inputs: inputs are gross complements when $\sigma_i < 1$, gross substitutes when $\sigma_i > 1$, and Cobb–Douglas in the limit as $\sigma_i \to 1$. We assume inelastic labor supply and normalize $L_{i,t} = 1$. The propagation analysis focuses on intermediate-input linkages.

For each customer firm i, we impose the row-normalization

$$\sum_{j=1}^{N} \omega_{ij,t} = 1, \tag{1}$$

where $\omega_{ij,t}$ denotes the primitive importance of supplier good j in firm i's intermediate-input bundle. We collect these primitive input weights in

$$W_t \equiv [\omega_{ij,t}]_{i,j=1}^{N}.$$

Row i of W_t is the vector of primitive input weights for customer firm i. These weights sum to one by the normalization condition in Equation (1). Column j collects the primitive weights that all customer firms assign to supplier good j.

We focus on downside idiosyncratic productivity shocks and define the positive shock magnitude

$$u_{j,t} \equiv -\Delta z_{j,t} \geq 0.$$

A larger value of $u_{j,t}$ represents a larger deterioration in supplier firm j's productivity.

Let $P_{i,t}$ denote the price of firm i's output, and let $p_{i,t} \equiv \log P_{i,t}$. Under perfect competition and cost

minimization, the relative log price of good i satisfies

$$p_{i,t} = -z_{i,t} + \frac{\alpha_i}{1-\sigma_i} \log\left(\sum_{j=1}^{N} \omega_{ij,t} \exp\{(1-\sigma_i)p_{j,t}\}\right). \tag{2}$$

Equation (2) links firm i's equilibrium output price to its own productivity and to the prices of the supplier goods used in its intermediate-input bundle. Across the N firms, the recursion defines a system of N nonlinear equations in the N equilibrium prices $\{p_{i,t}\}_{i=1}^{N}$. The response of one firm's price to another firm's price depends on the full input-output network, because a price change affects direct customers and can then propagate through customers of customers. In the Cobb–Douglas limit, $\sigma_i \to 1$, the CES price system becomes linear. Appendix A.1 derives the price recursion and the associated local network multiplier.

We define the endogenous intermediate-input expenditure share of customer firm i on supplier input j as

$$s_{ij,t} \equiv \frac{\omega_{ij,t} \exp\{(1-\sigma_i)p_{j,t}\}}{\sum_{k=1}^{N} \omega_{ik,t} \exp\{(1-\sigma_i)p_{k,t}\}}. \tag{3}$$

The object $s_{ij,t}$ is the share of firm i's intermediate-input expenditure paid to supplier firm j. Its response to a change in supplier price depends on firm i's elasticity of substitution across intermediate inputs.

Suppose supplier firm j experiences a negative productivity shock. In firm j's own pricing equation, the term $-z_{j,t}$ rises. In the competitive unit-cost formulation, this increase raises firm j's marginal cost and therefore raises its equilibrium output price $p_{j,t}$, holding other input prices fixed. Equivalently, the shock can be interpreted as a reduction in the availability of input j, which raises the shadow cost of using that input in downstream production. The higher price or shadow cost of good j then enters the CES input-cost aggregator of every customer firm i with $\omega_{ij,t} > 0$.

If $\sigma_i < 1$, inputs are gross complements for customer firm i. In this case, $1 - \sigma_i > 0$, and Equation (3) implies that an increase in $p_{j,t}$ raises $s_{ij,t}$. The customer firm cannot easily substitute away from the disrupted input, so the expensive or scarce input absorbs a larger expenditure share and becomes a bottleneck. If $\sigma_i > 1$, inputs are gross substitutes for customer firm i. In this case, $1 - \sigma_i < 0$, and an increase in $p_{j,t}$ lowers $s_{ij,t}$. The customer firm substitutes away from the expensive or scarce input, which attenuates the shock. Thus, the same primitive input weight $\omega_{ij,t}$ can generate different propagation strength depending on the customer's substitution elasticity σ_i and on equilibrium input prices.

We define the local price-pass-through matrix $B_t = [B_{ij,t}]_{i,j=1}^N$ by

$$B_{ij,t} \equiv \alpha_i s_{ij,t}.$$

The element $B_{ij,t}$ combines two objects: the share of firm i's intermediate-input expenditure paid to supplier firm j, and the intermediate-input share in firm i's production. Thus, $B_{ij,t}$ measures the local sensitivity of firm i's log unit cost to the log price of supplier good j.

Differentiating the CES price system around the equilibrium gives the local propagation equation

$$\Delta p_t = -\Delta z_t + B_t \Delta p_t.$$

If the spectral radius of B_t is smaller than one, this system has the solution

$$\Delta p_t = -\mathcal{L}_t \Delta z_t, \qquad \mathcal{L}_t \equiv [\ell_{ij,t}]_{i,j=1}^N = (I - B_t)^{-1} = \sum_{m=0}^{\infty} B_t^m. \tag{4}$$

The matrix $\mathcal{L}_t$ is the network multiplier used in this paper. It summarizes the first-order and higher-order pass-through of firm-level productivity shocks through the realized input-expenditure network. The identity matrix in the series captures the direct effect of a firm's own productivity shock on its own price. The term B_t captures the first-round effect through direct input-output links. The powers $B_t^2, B_t^3, \ldots$ capture repeated pass-through through longer production paths, including paths that involve network cycles.[5] Therefore, $\mathcal{L}_t$ is a multiplier of the equilibrium expenditure-share network. The element $\ell_{ij,t}$ measures the full pass-through of a productivity shock originating at supplier firm j to customer firm i through both direct and higher-order production paths. In the Cobb–Douglas limit, $s_{ij,t} = \omega_{ij,t}$, so $\mathcal{L}_t$ coincides algebraically with the standard Leontief inverse. In the general CES case, however, $s_{ij,t}$ depends on primitive input weights, substitution

[5] It is useful to distinguish $\mathcal{L}_t$ from the standard Leontief inverse in production-based multisector models. With the row convention used here, a standard direct-requirements matrix can be written as

$$C_{ij,t} \equiv \alpha_i \omega_{ij,t},$$

where $\omega_{ij,t}$ is the primitive input weight of supplier good j in customer firm i's production. The corresponding standard Leontief inverse is

$$\mathcal{L}_t^{standard} \equiv (I - C_t)^{-1}.$$

This object is built from primitive input requirements and is commonly used to map shocks or final-demand changes into gross-output, sales, or Domar-weight responses, depending on the specific production environment. By contrast, the multiplier used in this paper is defined in Equation (4) and is built from $B_{ij,t} = \alpha_i s_{ij,t}$, where $s_{ij,t}$ is the endogenous CES expenditure share. Hence, $\mathcal{L}_t$ is not, in general, the primitive Leontief inverse.

elasticities, and equilibrium prices, so $\mathcal{L}_t$ is an endogenous propagation multiplier rather than a primitive direct-requirements inverse.

2.2 Default Spillovers

We use the Merton distance-to-default framework to map firm fundamentals into default risk. We assume that firm i has asset value $V_{i,t}$, asset volatility $\sigma_{V,i,t}$, and zero-coupon debt with face value F_i maturing at T. Default occurs at maturity if $V_{i,T} < F_i$. Under lognormal asset dynamics, the risk-neutral distance to default is

$$DD_{i,t} = \frac{\ln\left(V_{i,t}/F_i\right) + \left(r - \frac{1}{2}\sigma_{V,i,t}^2\right)(T-t)}{\sigma_{V,i,t}\sqrt{T-t}},$$

where r is the risk-free rate. A lower asset value reduces the numerator of $DD_{i,t}$. A higher asset volatility increases the denominator and lowers the drift-adjusted numerator. Both changes reduce distance to default.

Asset value and asset volatility are not directly observed. We follow Bharath and Shumway (2008) and approximate them using equity value, debt face value, and equity volatility:

$$V_{i,t} \approx E_{i,t} + F_i, \tag{5}$$

$$\sigma_{V,i,t} \approx \frac{E_{i,t}}{V_{i,t}}\sigma_{E,i,t}. \tag{6}$$

In the production economy, we treat equity value as supported by operating sales revenue and cash flow. The approximations in Equations (5)–(6) map equity value and equity volatility into the Merton distance-to-default formula.

The risk-neutral default probability over $[t, T]$ is approximated by

$$\mathbb{P}_{i,t}^{\mathbb{Q}}(\text{default by } T) \approx \Phi(-DD_{i,t}), \tag{7}$$

and, with fractional recovery of par $R \in (0, 1)$ and a zero-coupon CDS approximation, the par CDS spread is

$$s_{i,t}^{CDS} = \frac{1-R}{T-t}\Phi(-DD_{i,t}). \tag{8}$$

Thus, any shock that lowers distance to default raises both risk-neutral default probability and the CDS spread under this approximation. Appendix A.2 reports the fully substituted nonlinear mapping from the

production-network exposures to distance to default and CDS spreads.

2.3 Model Implications

We now state the three main model implications. The first implication formalizes how the full network structure, featuring both direct and indirect input-output relationships, prices CDS spreads through both the cash-flow level channel and the cash-flow volatility channel. The second implication formalizes the importance of network topology in transmitting finite idiosyncratic shocks to downstream actual default. The third implication formalizes the importance of substitution elasticity in transmitting an infinitely large productivity shock, which directly drives the shocked firm into default, into an economy-wide default event.

Implication 1 (Full network structure prices credit spreads nonlinearly)**.** *For each firm i, distance to default is a nonlinear function of the full network multiplier $\mathcal{L}_t = [\ell_{ij,t}]_{i,j=1}^{N}$:*

$$DD_{i,t} = \mathcal{D}_i\left(\mathcal{L}_t; \bar{E}_{i,t}, \bar{\sigma}_{E,i,t}, F_i, r, T-t\right). \tag{9}$$

Substituting Equation (9) into the CDS pricing approximation in Equation (8) gives

$$s_{i,t}^{CDS} = \frac{1-R}{T-t}\Phi\left[-\mathcal{D}_i\left(\mathcal{L}_t; \bar{E}_{i,t}, \bar{\sigma}_{E,i,t}, F_i, r, T-t\right)\right]. \tag{10}$$

Therefore, orthogonal negative idiosyncratic productivity shocks propagate through both direct and indirect input-output relationships and enter the pricing of the cross section of CDS spreads through the full network multiplier.

The economic mechanism behind Implication 1 follows from the mapping from production-network shocks to firm fundamentals. Credit spreads are nonlinear functions of asset value and asset volatility. By the Bharath–Shumway approximations in Equations (5)–(6), asset value and asset volatility are functions of equity value and equity volatility. The production network therefore affects CDS spreads through the way it changes equity value and equity volatility.

For each firm i, equity value and equity volatility can be characterized in reduced form as

$$E_{i,t} = \bar{E}_{i,t} - \kappa^E_{i,t} X^L_{i,t}, \qquad X^L_{i,t} \equiv \sum_{j=1}^{N} \ell_{ij,t} u_{j,t}, \tag{11}$$

$$\sigma_{E,i,t} = \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} X^V_{i,t}\right)^{1/2}, \qquad X^V_{i,t} \equiv \sum_{j=1}^{N} v_{ij,t} = \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}, \tag{12}$$

where $\bar{E}_{i,t}$ and $\bar{\sigma}^2_{E,i,t}$ are equity value and equity variance absent the supplier-network shock, $\kappa^E_{i,t} > 0$ maps cash-flow losses into equity-value losses, and $\kappa^{\sigma}_{i,t} \geq 0$ maps network cash-flow variance into equity variance.

Equations (11)–(12) show that the network multiplier enters firm fundamentals through two channels. First, a negative supplier shock reduces customer operating cash flow in proportion to $\ell_{ij,t}$, which measures the customer's total direct and indirect network exposure to that supplier. Second, under conditional orthogonality of shocks, supplier j's contribution to customer i's cash-flow volatility equals the supplier's shock variance multiplied by the square of the network multiplier, $\ell^2_{ij,t}\sigma^2_{u,j,t}$. By Equations (7) and (8), both channels reduce distance to default and raise customer CDS spreads under the risk-neutral measure.

Thus, as illustrated by Implication 1, a firm-specific productivity shock can affect the entire cross section of credit spreads through the network multiplier. A negative productivity shock to a supplier first reduces the supplier's own output and cash flow, lowering the supplier's own distance to default. The same shock then transmits to other firms, directly and indirectly, through higher input costs. These higher input costs reflect both a higher equilibrium price of the upstream good and a higher shadow price associated with input scarcity. The resulting decline in customer operating cash flow lowers customer distance to default and raises customer CDS spreads.

Proof. See Appendix A.3. □

We next examine how network topology determines the severity of risk spillovers. We show that a hub-and-spoke structure, in which a dominant supplier provides a large share of inputs to many customer firms, is especially fragile because the dominant supplier's shock is transmitted broadly through the production network. We formalize this structure in the following assumption and implication.

Assumption 1 (Central supplier)**.** *Supplier firm c is central in the production network if, for some $\varepsilon \in [0, 1)$*

and a nonempty set of customer firms $\mathcal{I} \subseteq \{1, \ldots, N\}$,

$$s_{ic,t} \geq 1 - \varepsilon \qquad \text{for all } i \in \mathcal{I},$$

and the supplier uses its own output as an important input,

$$s_{cc,t} \geq 1 - \varepsilon.$$

The first condition states that supplier c accounts for a large expenditure share for many customer firms. The second condition states that supplier c has a strong self-input feedback loop. Since $B_{cc,t} = \alpha_c s_{cc,t}$*, where* α_c *is the intermediate-input share in firm c's production, a near-bottleneck supplier has* $\alpha_c(1 - \varepsilon)$ *close to one.*

Implication 2 (Network topology and the severity of finite-shock spillovers)**.** *Consider a finite downside productivity shock* $u_{c,t} > 0$ *to supplier firm c. For any target probability* $q \in (0, 1)$*, there exists a degree of centrality such that the finite supplier shock leaves the supplier solvent but makes the customer firm's risk-neutral default probability at least q:*

$$DD_{c,t}(u_{c,t}) > \bar{d}_c \quad \text{and} \quad \mathbb{P}^{\mathbb{Q}}_{i,t}(\text{default by } T \mid u_{c,t}) \geq q. \tag{13}$$

In the limiting bottleneck case $\alpha_c(1 - \varepsilon) \uparrow 1$ *in Assumption 1, the customer firm's default probability converges to one:*

$$\lim_{\alpha_c(1-\varepsilon)\uparrow 1} \mathbb{P}^{\mathbb{Q}}_{i,t}(\text{default by } T \mid u_{c,t}) = 1, \tag{14}$$

even though the supplier shock is finite and the supplier itself remains solvent by assumption.

Implication 2 shows that a finite supplier shock can propagate through the economy and move customer firms closer to the default boundary. When the supplier is the dominant input source for many firms and also uses its own output as an important input, the network multiplier is strongest. In this hub-and-spoke structure, even a finite shock that only moves the supplier closer to default, while leaving it solvent, can push customer firms into actual default. The reason is that customer firms rely heavily on the supplier. Even a moderate increase in the cost or scarcity of the supplier's product can prevent customer firms from obtaining the inputs needed to complete production, leading to bankruptcy.

The same topology can also generate feedback from the periphery back to the hub. If the central supplier sources only a small share of inputs from peripheral firms but relies heavily on its own output as an intermediate input, then default among peripheral firms can feed back to the supplier through those thin input links and ultimately weaken the supplier itself. When the outside inputs are complementary, the unavailability of those inputs can prevent the supplier from completing production. When the inputs are substitutable, the supplier reallocates toward its own output, which strengthens the self-input feedback loop. In both cases, the negative idiosyncratic shock can be amplified by the hub-and-spoke topology and can destroy production.

Therefore, Implication 2 shows that network topology is an important determinant of the severity of downside risk spillovers. Even when the supplier initially suffers only a finite shock, the shock can lead customer firms to actual default once the supplier becomes the dominant hub of production in the economy.

Proof. See Appendix A.4. □

Next, we formalize the importance of substitution elasticity in transmitting actual default risk across the economy.

Implication 3 (Elasticity of substitution and tail default propagation)**.** *Consider an extreme supply disruption, that is, an infinitely large negative productivity shock to supplier firm c that directly pushes the supplier into actual default:*

$$z_t = -xe_c, \qquad x \to \infty,$$

where $x > 0$ is the shock size, and e_c is the $N \times 1$ unit vector with one in position c and zero elsewhere. Equilibrium prices satisfy the asymptotic representation

$$p_{i,t}(-xe_c) = \mu_i^c + \phi_i^c x + o(x), \qquad x \to \infty,$$

where ϕ_i^c is firm i's tail exposure to a large negative shock in supplier firm c. The tail exposures solve

$$\phi_i^c = \mathbf{1}\{i = c\} + \alpha_i \begin{cases} \max_{j \in \mathcal{S}_i} \phi_j^c, & \sigma_i < 1, \\ \sum_{j \in \mathcal{S}_i} \omega_{ij} \phi_j^c, & \sigma_i = 1, \\ \min_{j \in \mathcal{S}_i} \phi_j^c, & \sigma_i > 1, \end{cases} \tag{15}$$

where $\mathcal{S}_i \equiv \{j : \omega_{ij} > 0\}$ is the set of inputs used by firm i.

If $\phi_i^c > 0$, then the default risk of supplier firm c drives firm i's distance to default to negative infinity and its risk-neutral default probability to one:

$$\lim_{x \to \infty} DD_{i,t}(-xe_c) = -\infty, \qquad \lim_{x \to \infty} \mathbb{P}_{i,t}^{\mathbb{Q}}(\textit{default by } T \mid z_t = -xe_c) = 1. \tag{16}$$

Implication 3 states that, in the case of an actual supplier default caused by an extremely negative productivity shock, the elasticity of substitution determines whether the supplier's default becomes an economy-wide credit-risk event. Complementary production links transmit the shock downstream. Substitutable production links can block the shock because customer firms can reallocate away from the disrupted input.

The intuition follows from the expenditure share $s_{ij,t}$. When $\sigma_i < 1$, an increase in the disrupted input price raises the expenditure share on that input because the customer firm cannot easily replace it. In the tail, the disrupted input becomes the bottleneck and the CES recursion converges to a max operator as in Equation (15). When $\sigma_i > 1$, the same price increase lowers the expenditure share because the customer firm substitutes toward less affected inputs. In the tail, the CES recursion converges to a min operator, and an unaffected alternative input can block propagation.

While the previous implication shows that detailed input-output structure matters for finite downside-shock transmission, this implication shows that the leading tail propagation rule is governed by whether downstream inputs are complementary or substitutable, rather than by small differences in finite input weights.

Proof. See Appendix A.5. □

3 Empirical Evidence

We provide empirical support for the structural framework by testing whether network topology improves the prediction of credit spreads. We use graph neural networks (GNNs) to conduct CDS-spread prediction exercises. GNNs are designed for data observed on graphs and can incorporate both node characteristics and edge weights, which makes them suitable for implementing the network exposures implied by the structural framework.

The GNN architecture represents the supply-chain spillover network explicitly. Each firm is modeled as a node, while directed edge weights are inherited from the industry-level spillover matrix according to the industries of the two firms. Thus, the edge from firm j to firm i is interpreted as the exposure of firm i's industry to disruption originating in firm j's industry; it is not interpreted as a direct bilateral product-market rivalry relation. The model takes both node-level features—such as firm-specific characteristics and macroeconomic variables—and edge-level features—namely industry-level idiosyncratic risk spillover measures imposed on firms by industry classification—as inputs. Through message passing and aggregation, the GNN incorporates supply-chain topology into firm-level latent representations, which are then used to predict CDS spreads.

We briefly describe the GNN architecture below and relegate architecture details to the Internet Appendix. We then describe the data construction, with particular emphasis on the measurement of node-level and industry-level edge features, and finally present the CDS spread prediction results.

3.1 Architecture of the Empirical Algorithm

Much of the machine-learning literature emphasizes extensive model tuning and increasingly sophisticated architectures to maximize out-of-sample predictive performance. Our objective and approach differ. We deliberately employ a parsimonious vanilla GNN architecture that is sufficient to deliver a clear financial interpretation of network effects.

The graph neural network (GNN) framework has two components: an inter-layer message-passing scheme and an intra-layer updating scheme. The intra-layer updating scheme corresponds to the standard transformation used in conventional neural-network architectures such as CNNs.

Figure 1 illustrates the inter-layer message-passing scheme that is specific to GNNs.

FIGURE 1 ABOUT HERE

We first define the objects in the message-passing equation. Let $h_i^{(k)}$ denote the embedding of firm node i at layer k, let $\hat{h}_i^{(k)}$ denote the edge-augmented embedding, let $\mathcal{N}(i)$ denote the set of neighbors of node i, let a_{ij} denote the directed spillover weight from node j to node i, and let $\bigoplus$ denote concatenation. The scheme specifies how node-level features and edge weights enter latent node representations. Specifically, each normalized embedding $\hat{h}_i^{(k)}$ at layer k incorporates information from node i and its neighbors according to

$$\hat{h}_i^{(k)} = h_i^{(k)} \bigoplus_{j \in \mathcal{N}(i)} (a_{ij}), \tag{17}$$

where $\hat{h}_i^{(k)}$ denotes the normalized embedding of node i at layer k, $h_i^{(k)}$ is the embedding of node i at layer k, $\bigoplus$ represents the concatenation operator, and $\mathcal{N}(i)$ denotes the neighborhood of node i.

In Equation (17), a_{ij} is the directed idiosyncratic risk spillover from node j to node i. Collecting all a_{ij} yields the adjacency matrix $A \equiv [a_{ij}]$. To encode network effects, we follow the graph convolutional network (GCN) formulation of Kipf and Welling (2016). Before writing the graph-convolution equation, let $H^{(k)}$ be the matrix of node embeddings entering layer k, let $W^{(k)}$ be the trainable weight matrix, let $\hat{A} = A + I$ be the adjacency matrix with self-loops, and let D be the diagonal degree matrix used for normalization. The graph-convolution transformation is

$$\hat{H}^{(k)} = D^{-1}\hat{A}H^{(k)}W^{(k)}, \quad \text{if } A \text{ is asymmetric}, \tag{18}$$

where $\hat{H}^{(k)}$ is the output of the k-th layer, $\hat{A} = A + I$ with I denoting the identity matrix, D is the out-degree matrix, $H^{(k)}$ is the input to the k-th layer, and $W^{(k)}$ is the trainable weight matrix. The out-degree matrix D is diagonal, with diagonal elements equal to the column sums of A.

Equation (18) states the graph-convolution transformation used to incorporate the adjacency matrix into node representations.

The economic interpretation is as follows. Row i of the adjacency matrix contains the spillover weights from supplier industries to firm i's industry. Multiplying this row by neighboring firms' features gives greater weight to firms located in supplier industries with larger spillover weights. After aggregating weighted supplier-industry information together with firm i's own characteristics, the GNN constructs a latent representation of firm i. The matrix D^{-1} serves to normalize the adjacency matrix. The contribution

of neighboring information therefore depends on the measured spillover weight from the supplier industry to firm i's industry.

The intra-layer updating scheme governs how node representations are transformed from one hidden layer to the next. This structure is standard across neural network architectures. Each layer applies a sequence of operations to the node representations, including batch normalization, dropout, nonlinear activation (ReLU), and an aggregation function. In standard neural networks, the aggregation function typically takes the form of an equally weighted average. In GNNs, however, aggregation is performed by combining neighboring node characteristics using edge-weighted averages, thereby reflecting the relative importance of neighboring information.

The architecture described above corresponds to the vanilla GNN baseline, which is fundamentally cross-sectional. That is, it operates on a single cross-sectional network snapshot to predict outcomes in the subsequent period. Unlike CNNs, which naturally incorporate temporal pooling, the vanilla GNN does not include an explicit mechanism for temporal aggregation. Although we implement an extended architecture with pooling over time in the Internet Appendix, we focus on the baseline specification in the main analysis. We use this specification for two reasons. First, it is consistent with our structural framework, which emphasizes cross-sectional network effects rather than explicit dynamic propagation. Second, introducing temporal pooling requires additional assumptions—such as recurrent structures (e.g., LSTM)—that increase model complexity and reduce interpretability. Moreover, CDS data are sparse and unevenly observed across firms and time, which necessitates masking schemes that further obscure economic interpretation. Therefore, we focus on the vanilla structure in the main analysis.

3.2 Data and Implementation Details

We use daily Markit CDS data for U.S. firms spanning January 2005 to December 2020.[6] We focus on the 5-year tenor and senior unsecured contracts, which are the most liquid in the CDS market. We use spreads from the prevalent XR14 contract. The spread on XR contracts reflects default risk while excluding restructuring risk, which aligns with our structural framework.[7]

We construct a monthly panel by retaining the most recent CDS spread observed in each month. The

[6] Prior to 2005, CDS coverage is limited, rendering the data less suitable for machine-learning applications.

[7] As documented by Liu (2022), XR contracts became the standard for U.S. corporates following the 2009 CDS Big Bang, whereas MR (modified restructuring) contracts were more common prior to that event. XR spreads primarily reflect default risk, while MR spreads reflect both default and restructuring risk.

final dataset contains 678 firms and 130,176 firm-month observations over the sample period.

Table 1 reports summary statistics for the distribution of log CDS spreads, along with firms' market capitalization and implied credit ratings. Columns 1–2 show that CDS contracts are observed for a median duration of 137 months (11.4 years), with a minimum of 2 months and a maximum of 192 months (16 years). Columns 3–4 indicate that the sample spans firms ranging from very small (market capitalization of $700 million) to very large (market capitalization of $1.73 trillion). Columns 5–6 show that firms cover a wide spectrum of credit quality, from AA-rated to CCC- and D-rated firms.[8] These statistics highlight the broad coverage of the CDS market in terms of firm size, credit quality, and sample duration.

We use the logarithm of CDS spreads as the target variable. CDS spreads are strongly right-skewed, and commonly used loss functions in machine learning, such as the mean squared error (MSE), are highly sensitive to skewness: large observations receive disproportionate weight and can dominate the loss. As a result, a model trained on levels may overfit extreme observations at the expense of fitting the bulk of the distribution. To mitigate this issue, we apply a logarithmic transformation to CDS spreads and use the transformed variable as the target in our prediction model. Figure 3 presents the histogram of log CDS spreads. The mean log CDS spread is approximately −4.69, with a standard deviation of 0.94.

Node Characteristics. We construct 94 firm-level characteristics following Gu, Kelly, and Xiu (2020). Of these, 61 characteristics are updated annually, 13 quarterly, and 20 monthly. These characteristics are designed to capture distinct and largely nonredundant information about firm fundamentals.[9]

In addition, we include eight macroeconomic variables from Welch and Goyal (2008): the dividend-price ratio (dp), earnings-price ratio (ep), book-to-market ratio (bm), net equity issuance (ntis), Treasury-bill rate (tbl), term spread (tms), default spread (dfy), and stock variance (svar).[10] In total, we use 112 node-level features. Following Kelly, Pruitt, and Su (2019) and Freyberger, Neuhierl, and Weber (2020), all characteristics are rank-normalized cross-sectionally each month and mapped to the interval $[-1, 1]$.

[8] The firm count exceeds 678 because some firms experience rating changes over time and are therefore associated with multiple ratings during the sample period. Sectoral distributions of CDS contracts are reported in Internet Appendix B.

[9] We account for data release delays following Gu, Kelly, and Xiu (2020) and Gu, Kelly, and Xiu (2021). The underlying data and replication codes for early years are provided by the authors. Related literature includes Fama and French (2016), Green, Hand, and Zhang (2017), Hou, Xue, and Zhang (2020), Gu, Kelly, and Xiu (2021), and Kelly, Malamud, and Zhou (2024). Details of all characteristics are provided in Internet Appendix C.

[10] Including interactions between firm-level and macroeconomic variables yields similar out-of-sample results.

Edge Characteristics. To construct edge-level features, we estimate pairwise idiosyncratic volatility spillovers using CRSP daily stock returns. The estimation is performed at the industry level, consistent with the model's supply-chain interpretation. We assume that firms within the same industry produce similar products and are exposed to similar productivity-specific and supply-chain shocks. Accordingly, the empirical network should be interpreted as an industry-level supply-chain spillover network rather than a firm-level product-market rivalry network.

Specifically, we compute daily industry returns as value-weighted averages of firm returns. For each calendar month, we regress daily industry returns on the Fama–French three factors and treat the residuals as idiosyncratic industry returns.[11] Monthly idiosyncratic volatility is then computed as the standard deviation of daily idiosyncratic returns.

We estimate pairwise idiosyncratic volatility spillovers using a rolling-window LASSO vector autoregression (VAR). For each 90-month rolling window, we estimate a VAR for the panel of sector-level log idiosyncratic volatilities and perform a generalized variance decomposition (GVD) for 6-month-ahead forecast errors.[12] This procedure yields a 48×48 matrix of spillover intensities. Element $a_{gh,t}$ measures the contribution of shocks originating in supplier industry h to the idiosyncratic volatility of downstream industry g. These measures constitute the empirical counterpart of the structural industry-level spillover weights in the model.

We collect the estimated $a_{gh,t}$ from each rolling window to form a time-varying adjacency matrix A_t updated at a monthly frequency.[13]

Extrapolation to the Firm Level. After obtaining industry-level spillover measures, we extrapolate them to the firm level by imposing industry-level supply-chain spillovers on firms according to their industry classifications. Figure 2 illustrates this procedure.

FIGURE 2 ABOUT HERE

Figure 2 shows an example with three sectors and five firms. If the spillover intensity from sector 2 to sector 1 equals 0.1, then this value is assigned as the spillover exposure from any firm in sector 2 to any firm in sector 1.

[11] Results are robust to removing only the CAPM factor, removing five principal components, or not removing common factors at all.

[12] Results are robust to rolling windows ranging from 80 to 100 months and forecast horizons between 6 and 10 months.

[13] The adjacency matrix at time t is constructed using data from $t-90$ to t. Results are robust to lagging edge inputs by one month to avoid look-ahead bias. Prior work (e.g., Kryzanowski, Perrakis, and Zhong, 2017) finds that CDS markets tend to lead equity markets in response to negative news by days or weeks.

By imposing sector-level adjacency matrices at the firm level, we map industry-level supply-chain spillovers into firm-level prediction inputs. This approach assumes that firms within the same sector are representative of sector-level production and risk exposure. While heterogeneity undoubtedly exists within industries, this assumption is consistent with the paper's focus on industry-level supply-chain disruption. For clarity, we abstract from within-industry product-market rivalry and additional firm-specific shock channels.

Finally, one may ask why we do not construct idiosyncratic volatility spillovers directly from CDS data. We do not pursue this approach because corporate CDS data exhibit substantial noise, sparse coverage, infrequent trading, and non-transaction-based quotes that vary significantly across time and firms. In contrast, equity market data are cleaner, higher frequency, and better suited for constructing reliable spillover measures. Moreover, using stock data aligns naturally with our structural framework, in which equity prices reflect firm value and default risk.

Training Configuration and Benchmark We construct a monthly panel in which the inputs consist of firm-level node characteristics and inter-firm edge characteristics, and the target variable is the CDS spread. As discussed above, we adopt a vanilla GNN architecture that operates primarily in the cross section. For each month t, we train the GNN using the full cross-sectional network observed at month t. We then use the $t + 1$ snapshot as a validation set to tune hyperparameters and prevent overfitting, and we generate out-of-sample predictions for month $t + 1$. For all GNN specifications, we use stochastic gradient descent (SGD) as the optimizer. To mitigate overfitting, we implement early stopping and select all hyperparameters by minimizing the mean squared error on the validation set.

We recursively refit the model each month to incorporate the most recent node-level and edge-level information, despite the associated computational cost, until the end of the sample period. This rolling training scheme ensures that all reported predictions are strictly out of sample.

To benchmark the performance of models that do not incorporate network edge information, we consider a set of alternative nonlinear machine-learning methods, including convolutional neural networks (CNN), gradient boosting regression trees (GBRT), random forests (RF), principal component regression (PCR), partial least squares (PLS), and support vector regression (SVR). These models use the same set of firm-level node characteristics as inputs but do not incorporate inter-firm edge features. All competing models are trained using the same rolling data partitions and evaluated under the same out-of-sample protocol as the GNN.

To provide a clean and internally consistent benchmark for the GNN, the CNN model is designed under strict architectural restrictions such that it is equivalent to a GNN with all network edges removed. Specifically, this benchmark preserves the same inter-layer and intra-layer structure as the GNN, but replaces the adjacency matrix with the identity matrix, effectively eliminating all cross-node message passing. The model uses the same number of layers, activation functions, and parameter-sharing structure as the GNN, and excludes any graph-level normalization. We hand-code the architecture to ensure that this zero-edge GNN collapses exactly to a per-node feedforward network. For simplicity, we refer to this restricted model as CNN throughout the paper. Detailed descriptions of all comparison algorithms are provided in Internet Appendix D.

3.3 Empirical Results

We evaluate the importance of network edge features in explaining credit spreads by examining both pooled out-of-sample (OOS) performance across the entire sample and time-series variation in OOS performance. The OOS evaluation provides an objective assessment of predictive accuracy.

Pooled Out-of-Sample Performance. Table 2 reports pooled OOS performance for all machine-learning models. Panel A reports pooled out-of-sample root mean squared errors (RMSEs), while Figure 4 visualizes these results using bar plots.

Column 1 of Table 2 reports RMSEs for the full sample from March 2005 to December 2020. Columns 2 and 3 report RMSEs for investment-grade (BBB and above) and high-yield (BB and below) firms, respectively. Columns 4 and 5 report RMSEs for firms below and above the median market capitalization.

The results in Column 1 show that GNN models, across different architectural depths, achieve RMSEs of approximately 0.89—less than half of those produced by alternative algorithms. Under identical training conditions, CNN models exhibit substantially higher RMSE: 1.34 with two hidden layers. While increasing CNN depth initially improves performance, additional layers lead to deterioration, likely due to overfitting. In contrast, GNN performance is remarkably stable across architectures: even a single hidden layer is sufficient to capture the relevant network topology, and additional layers yield limited incremental gains. This stability reflects diminishing returns from repeated aggregation over a fixed network structure.

Tree-based methods (GBRT and RF) outperform dimension-reduction approaches (PCR and PLS) and support vector regression (SVR), but still underperform relative to CNNs and substantially underperform

relative to GNNs. Overall, these results indicate that incorporating network topology—specifically inter-firm idiosyncratic risk spillovers—accounts for a substantial share of the cross-sectional variation in CDS spreads.

Columns 2 and 3 show that GNN models deliver substantial performance gains for both investment-grade and high-yield firms. For investment-grade firms, RMSEs from CNN2 are approximately 1.4, whereas GNN RMSEs are around 0.9. For high-yield firms, RMSEs from alternative models are approximately 1.2, compared with about 0.85 for GNNs. These results indicate that network information improves predictive accuracy across credit qualities, with particularly pronounced gains for investment-grade firms. Intuitively, investment-grade firms tend to be larger, more central, and more interconnected within production and trade networks; as a result, their CDS spreads are especially sensitive to network-wide spillovers, which GNNs are well suited to capture.

Columns 4 and 5 show that GNN models outperform competing algorithms for both small and large firms. For small firms, RMSE declines from 1.30 (CNN2) to 0.80 (GNN2), while for large firms RMSE declines from 1.39 to 0.96. These improvements are of comparable magnitude. However, we note that firms in our sample are predominantly small to medium-sized, and therefore we refrain from drawing strong conclusions regarding size-based heterogeneity.

Economic Significance of Network Effects. To assess the economic magnitude of network effects, Panel B of Table 2 reports two additional measures.

The first measure is the *network-attributable spread change* (NSC), which captures the magnitude of the network-induced change in credit-spread levels relative to a node-only benchmark. Before defining it, let $\widehat{\log s}^{\text{GNN}}$ denote the predicted log CDS spread from the GNN, let $\widehat{\log s}^{\text{CNN}}$ denote the predicted log CDS spread from the node-only benchmark, and let $\log s$ denote the realized log CDS spread:

$$\text{NSC} \equiv \frac{\mathbb{E}\left[\left|\widehat{\log s}^{\text{GNN}} - \widehat{\log s}^{\text{CNN}}\right|\right]}{\left|\mathbb{E}[\log s]\right|}. \tag{19}$$

Equation (19) states the network-attributable spread-change measure. Here, $\widehat{\log s}^{\text{GNN}}$ denotes the predicted log CDS spread from GNN2, $\widehat{\log s}^{\text{CNN}}$ denotes the predicted log CDS spread from CNN2, and $\log s$ denotes the realized log CDS spread. The numerator measures the average magnitude of the network-induced *change* in the predicted log spread level when network information is incorporated, while the denominator normalizes this change by a typical log spread level. This normalization yields a scale-free statistic that

facilitates comparison across samples, industries, and time.

NSC can be interpreted as an average treatment–style effect relative to a node-only baseline, quantifying how much the inclusion of network information shifts the level of predicted credit spreads. Panel B shows that incorporating network edge features induces an average spread change of approximately 21.8% in the full sample. The network-induced change is larger for high-yield firms (30.1%) than for investment-grade firms (18.5%), and larger for small firms (26.6%) than for large firms (17.7%).

In addition to NSC, we report an incremental explanatory-power measure, denoted Inc_R^2. The numerator below is the GNN mean squared prediction error, and the denominator is the CNN mean squared prediction error:

$$Inc\_R^2 \equiv 1 - \frac{\mathbb{E}\left[(\widehat{\log s}^{\,\mathrm{GNN}} - \log s)^2\right]}{\mathbb{E}\left[(\widehat{\log s}^{\,\mathrm{CNN}} - \log s)^2\right]}. \tag{20}$$

Equation (20) states the incremental explanatory-power measure relative to the node-only benchmark. This statistic measures the fraction of node-only prediction error eliminated by incorporating network edge features.

As reported in Panel B, the average incremental R^2 is approximately 0.56, with particularly pronounced gains for investment-grade firms and for relatively smaller firms in our sample. Given the limited size coverage of our data, we refrain from drawing strong conclusions regarding size-related heterogeneity. Nevertheless, the results consistently indicate that network edge information is especially informative for explaining the credit spreads of investment-grade firms, which tend to be more central and embedded in complex trade and production relationships.

While network edge information also improves the prediction of credit spreads for high-yield firms, the associated gains in explanatory power are more modest than those observed for investment-grade firms. A plausible explanation is that high-yield firms are closer to default, and their credit risk is more strongly driven by firm-specific fundamentals and balance-sheet conditions. As a result, firm-level characteristics account for a larger proportion of the variation in their credit spreads, even though network information remains economically relevant.

Time-Series Variation in Out-of-Sample Performance. We next examine the time-series performance of network features. Figure 5 plots the monthly out-of-sample RMSEs for each algorithm.

FIGURE 5 ABOUT HERE

Figure 5 shows that GNN models consistently outperform all competing models over time. In certain periods, GNN RMSEs are as low as one-quarter of those produced by alternative algorithms, particularly relative to dimension-reduction methods such as PCR and PLS. This persistent outperformance highlights the importance of pairwise industry-level supply-chain network information in explaining CDS spreads.

More importantly, the time-series results reveal distinct episodes during which network structure becomes especially informative for CDS pricing. The first such episode coincides with the 2008 Global Financial Crisis. During periods of severe economic stress, production networks are disrupted as firms face binding liquidity constraints, operational shutdowns, and heightened default risk. In this environment, inter-firm contagion and counterparty exposure become first-order determinants of credit spreads. As idiosyncratic shocks propagate through production and financial linkages, CDS spreads increasingly reflect network position rather than standalone firm fundamentals. GNNs are well suited to capture these contagion effects, whereas models that abstract from network structure perform poorly.

A second episode arises in late 2009 and early 2010, a transitional phase following the crisis. Although equity markets rebounded sharply during this period, production and counterparty networks remained impaired and were undergoing reconfiguration. Firms adjusted supplier relationships, renegotiated contracts, and reallocated production across newly constrained networks. This re-wiring process introduced substantial uncertainty that was not immediately reflected in balance-sheet or accounting data. As a result, models that rely solely on firm-level features exhibit a sharp deterioration in predictive performance. In contrast, GNNs continue to perform well by incorporating evolving inter-firm dependencies that shape downside credit risk during this network re-equilibration phase.

A third episode occurs around 2018, coinciding with the escalation of trade tensions and the introduction of tariff policies. From a network perspective, tariffs represent shocks to trade and production linkages that increase fragility within global value chains. Firms' exposure to affected suppliers and customers becomes a key determinant of credit risk, as disruptions propagate asymmetrically through the network. During this period, CDS spreads reflect not only firm-specific conditions but also indirect exposure to trade partners facing heightened uncertainty. The superior performance of GNNs during this episode underscores the role of network structure in transmitting trade-related shocks into credit markets.

Finally, following the onset of the COVID-19 pandemic in 2020, widespread production shutdowns generated severe upstream supply disruptions. The sudden closure of critical nodes—such as semiconductor manufacturing facilities—had cascading effects on downstream firms that depended on these inputs. Credit

risk during this period was therefore shaped by firms' positions within disrupted supply chains rather than by their pre-pandemic fundamentals alone. GNNs effectively capture these cascading network effects, while non-network models struggle to account for the resulting comovement in CDS spreads.

Taken together, the time-series evidence not only supports the predictions of our structural framework but also highlights the state-dependent importance of industry-level supply-chain networks in credit markets. In periods of stress, transition, or structural disruption, the full network topology captures an economically meaningful component of systematic risk that is priced in CDS spreads. These findings are consistent with the production-network literature (e.g., Acemoglu et al., 2012), which emphasizes that network structure provides a microfoundation for aggregate risk and its amplification.

To examine how global network features differentially affect sectors within the production network over time, we plot the network-attributable spread change (NSC) for firms across Fama–French 48 industries and across out-of-sample months.

Figure 6 presents a heatmap of the time-varying NSC across industries. For each month and each Fama–French 48 industry, we compute NSC as the average absolute difference between the predicted log CDS spread from GNN2 and that from CNN2, normalized by the average log CDS spread within the industry. At each month, industries are ranked by NSC. The five industries with the largest network-attributable spread change are highlighted in red, while the five industries with the smallest network-attributable spread change are highlighted in blue; all remaining industries are shown in white.

FIGURE 6 ABOUT HERE

As shown in Figure 6, several industries consistently appear among those with the highest NSC. These include Industry 22 (electrical equipment, covering electronic transmission and distribution equipment and electrical apparatus), Industry 25 (shipbuilding and railroad equipment), and Industry 26 (defense-related industries, including guided missiles and tanks).

In contrast, several industries consistently appear among those with the lowest NSC, including Industry 7 (entertainment, such as film and live performances), Industry 15 (rubber and plastic products), Industry 16 (textiles, including textile mill and canvas products), Industry 28 (mining, including metal and nonmetallic mining), and Industry 29 (coal, including bituminous coal and lignite mining).

The heatmap should be interpreted as identifying industries whose CDS spreads are most sensitive to network information. A high value of NSC means that adding supply-chain edge information changes predicted spreads substantially for firms in that industry.

The industries highlighted in red are consistent with the model's elasticity mechanism. Electrical equipment, shipbuilding and railroad equipment, and defense-related industries are plausibly sectors that rely on rigid complementary supplier relationships. Production in these industries often requires a set of specialized inputs to be jointly available and compatible with one another. Defense systems require specialized components, certified suppliers, precision electronics, engines, sensors, and materials; a single missing certified component can delay the entire product. Consistent with this interpretation, the defense-related sector appears especially network-sensitive during 2014–2018, a period that coincides with geopolitical shocks such as the Russia–Ukraine conflict and conflicts involving ISIS. During such periods, defense production and procurement conditions can become lumpy and state-dependent, and contracts may be renegotiated, accelerated, delayed, or repriced.

Electrical equipment also relies on specialized metals, semiconductors, power components, insulation materials, and precision parts. Many of these inputs are customized and difficult to replace quickly. Similarly, shipbuilding and railroad equipment are large system-integration industries in which inputs are not easily interchangeable. These sectors depend on engines, control systems, steel, navigation systems, propulsion systems, electrical systems, and other specialized components. Therefore, firms in these industries are likely to have low effective substitution elasticities and strong exposure to upstream supply-chain disruptions. This makes their CDS spreads especially sensitive to network information.

By contrast, the industries highlighted in blue are those for which network information changes predicted CDS spreads the least. These sectors are plausibly characterized either by more substitutable supplier relationships or by credit risk that is driven primarily by other elements. Textiles, for example, use inputs such as cotton, wool, silk, and synthetic fibers that are broadly available and globally substitutable, especially for standard fabrics. Rubber and plastic products also use many standardized petrochemical inputs, resins, and additives that can often be sourced from multiple suppliers.

Mining and coal are less dependent on highly specialized upstream inputs. Their main inputs include heavy machinery, such as drills and excavators, as well as labor, energy, maintenance services, and transportation services. However, credit risk in these sectors is often dominated by output-side fundamentals, including final demand, commodity prices, reserves, environmental regulation, and leverage, rather than by disruptions to specific suppliers.

Entertainment firms are similarly more exposed to demand, intellectual property, and consumer sentiment than to rigid upstream production bottlenecks. As a result, network edge information induces relatively

smaller shifts in predicted CDS spreads for these industries.

Overall, the industry heterogeneity evidence is most directly consistent with Implication 1: the full network structure affects the cross section of CDS spreads. It is also consistent with Implication 3: sectors with more rigid complementary supplier relationships exhibit larger network-attributable spread changes, while sectors with more substitutable suppliers or more standalone sources of credit risk exhibit smaller network-attributable spread changes.

4 Conclusion

This paper develops a structural framework for credit counterparty risk in which negative idiosyncratic productivity shocks propagate through a CES production network. Each firm is also a production sector: firm i produces good i and uses other firms' goods as intermediate inputs. The model separates the network effect into a cash-flow level channel and a cash-flow volatility channel. The high-order level exposure $\ell_{ij,t}$ maps supplier shocks into customer input costs, cash flows, and equity values. The quadratic exposure $\ell^2_{ij,t}\sigma^2_{u,j,t}$ maps supplier risk into customer equity volatility. Distance to default then transforms these two channels into a nonlinear credit-spread response.

The model delivers three implications. First, the full network structure prices the cross section of CDS spreads through both direct and indirect input-output relationships. Second, detailed network topology determines the severity of finite-shock spillovers: a hub-and-spoke structure with a dominant supplier can transmit a finite supplier shock broadly enough to push customer firms into actual default even when the supplier remains solvent. Third, in the tail case in which a supplier suffers an infinitely large productivity shock and defaults, the elasticity of substitution determines whether the disruption becomes an economy-wide credit-risk event. Complementary inputs propagate the shock because the disrupted input becomes a bottleneck, whereas substitutable inputs can attenuate or block propagation.

Empirically, we estimate industry-level network spillovers and impose them on firms according to industry classification. This implementation is consistent with the assumption that firms within the same industry produce similar goods and face similar productivity and supply-chain shocks. The paper deliberately abstracts from within-industry product-market rivalry and horizontal competition in order to isolate the production-network, or supply-chain, implication for default risk.

Using graph neural networks, we find that models incorporating network topology substantially out-

perform non-network benchmarks in predicting CDS spreads. The improvement is economically large, persistent across firm size and credit quality, and particularly pronounced during periods of financial stress, economic transition, and supply-chain disruption. Overall, the results suggest that CDS spreads reflect not only firms' standalone default risk, but also their exposure to shocks transmitted through the high-order structure of the production network.

Table 1. Summary Statistics of CDS Contracts and Issuing Firms

This table presents summary statistics for CDS contracts and their issuing firms from January 2005 to December 2020. Columns 1-2 show CDS contract duration statistics. Columns 3-4 show firms' market capitalization statistics. Columns 5-6 present the distribution of firms' credit ratings.

CDS Duration		Firm Market Capitalization		Firm Rating Distribution	
	Month		Dollar($)	Rating	Firm Count
mean	117	mean	2.65×10^7	AA	97
std	69	std	5.38×10^7	A	144
min	2	min	7.01×10^2	BBB	192
25%	50	25%	3.57×10^6	BB	156
50%	137	50%	1.01×10^7	B	88
75%	188	75%	2.64×10^7	CCC	59
max	192	max	1.71×10^9	D	3

Table 2. Out-of-Sample Performance and Network Attribution. Panel A reports pooled out-of-sample root mean squared errors (RMSEs) for alternative machine-learning models. Panel B reports the economic contribution of network edge information beyond node characteristics, measured by the network-attributable spread change (NSC) and the incremental R^2 (Inc_R^2). Column 1 reports the full sample (March 2005–December 2020). Columns 2–3 report investment-grade (BBB and above) and high-yield (BB and below) firms. Columns 4–5 report small (below-median market capitalization) and large (at or above median market capitalization) firms. NSC is defined as $\mathbb{E}[|\widehat{\log s}^{\text{GNN}} - \widehat{\log s}^{\text{CNN}}|]/|\mathbb{E}[\log s]|$. Inc_R^2 is defined as $1 - \text{MSE}_{\text{GNN}}/\text{MSE}_{\text{CNN}}$. Predicted log spreads are obtained from CNN2 and GNN2.

	All	Investment Grade	High Yield	Small	Big
Panel A: Pooled Out-of-Sample RMSE					
RF	1.665	1.828	1.271	1.464	1.845
GBRT	1.900	2.128	1.320	1.570	2.181
PCR	2.196	2.527	1.283	1.501	2.719
PLS	2.348	2.679	1.459	1.679	2.865
SVR	1.876	2.103	1.296	1.540	2.160
CNN1	2.245	2.308	2.110	2.072	2.405
CNN2	1.338	1.392	1.221	1.282	1.391
CNN3	1.378	1.437	1.249	1.337	1.418
GNN1	0.890	0.909	0.850	0.808	0.965
GNN2	0.888	0.904	0.854	0.809	0.960
GNN3	0.893	0.915	0.847	0.808	0.972
Panel B: Network-Attributable Spread Change and Incremental R^2					
NSC	0.218	0.185	0.301	0.266	0.177
Inc_R^2	0.560	0.578	0.511	0.602	0.524

Figure 1. Inter-Layer Design of the GNN
This figure depicts the inter-layer message-passing architecture of the GNN algorithm, where node embeddings are generated according to the GCN procedure proposed in Kipf and Welling (2016).

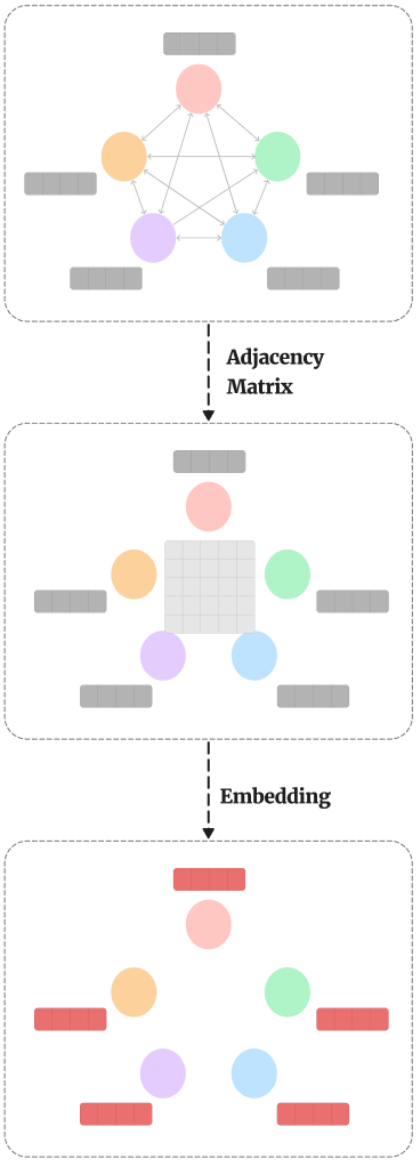
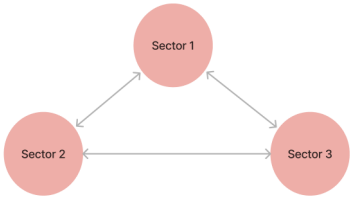


Figure 2. Extrapolation of Sector-Level Network Effects to Firm Level

This figure illustrates the process of extrapolating sector-level idiosyncratic risk spillover to the firm level. Panel A depicts risk spillover among three sectors, estimated from the Generalized Variance Decomposition (GVD) of VAR forecast errors. In the adjacency matrix, element (i, j) indicates the risk spillover intensity from sector j to sector i. Panel B demonstrates the extrapolation of this risk spillover to five firms across the three sectors.

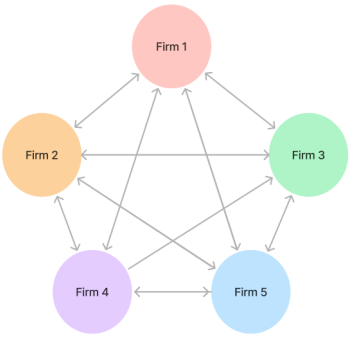
Panel A: Example for risk spillover among 3 sectors



	Sector 1	Sector 2	Sector 3
Sector 1	0.4	0.1	0.5
Sector 2	0.25	0.6	0.15
Sector 3	0.25	0.45	0.3

Panel B: Example for extrapolation

Firm	Sector 1	Sector 2	Sector 3
Firm 1	1	0	0
Firm 2	0	1	0
Firm 3	1	0	0
Firm 4	0	0	1
Firm 5	0	1	0



	Firm 1	Firm 2	Firm 3	Firm 4	Firm 5
Firm 1	0.4	0.1	0.4	0.5	0.1
Firm 2	0.25	0.6	0.25	0.15	0.6
Firm 3	0.4	0.1	0.4	0.5	0.1
Firm 4	0.25	0.45	0.25	0.3	0.45
Firm 5	0.25	0.6	0.25	0.15	0.6

Figure 3. Distribution of Log CDS Spreads
This figure displays the histogram of log CDS spreads for 5-year tenor, senior unsecured contracts from January 2005 to December 2020.

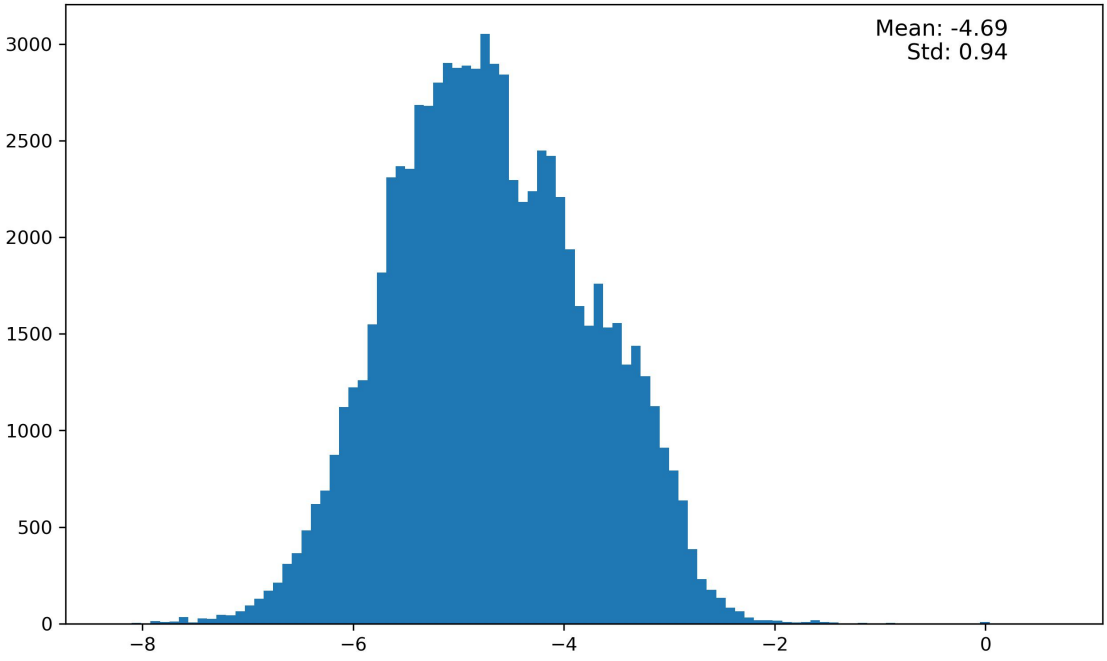
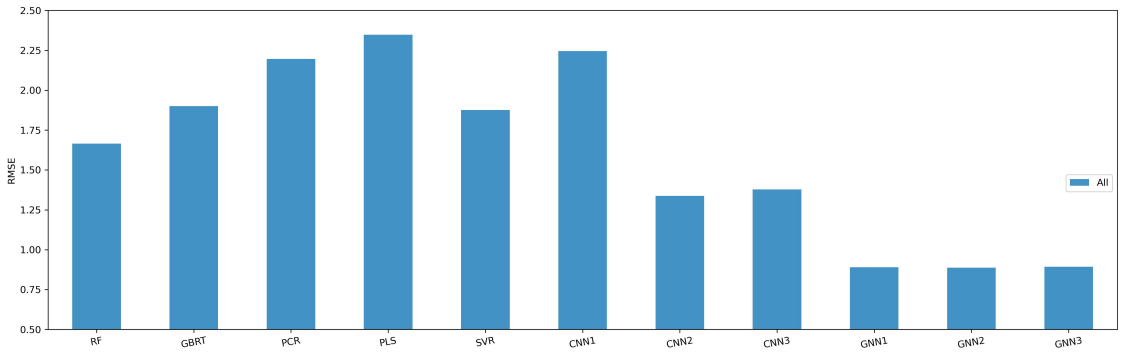
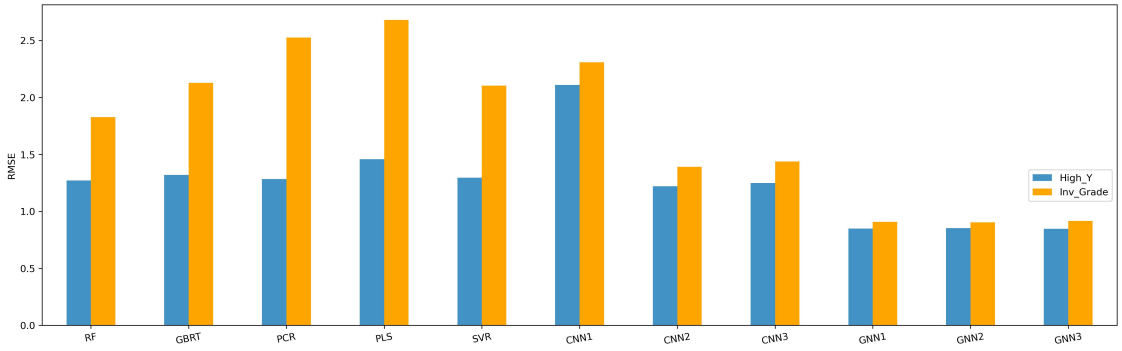


Figure 4. Out-of-Sample Prediction Error of Machine Learning Algorithms

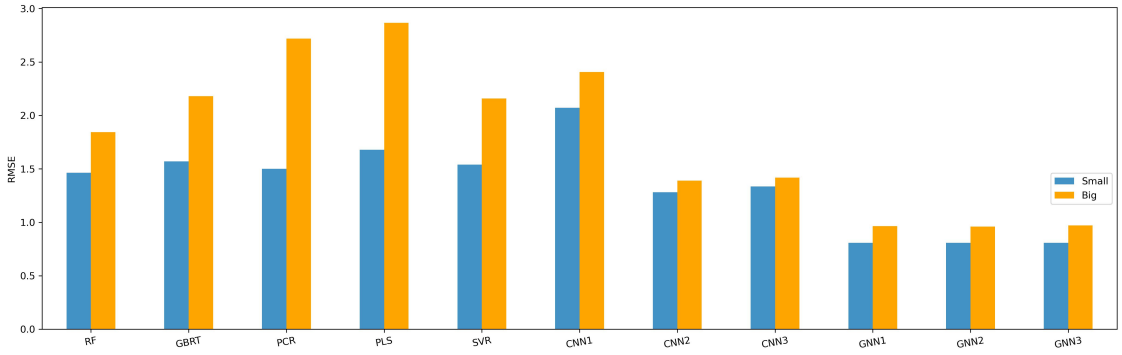
This figure presents the out-of-sample prediction error of various machine learning algorithms using pooled Root Mean Square Error (RMSE). Panel A displays the overall RMSE across all out-of-sample periods. Panel B presents the overall RMSE by firm credit rating, with investment grade defined as BBB or above and high-yield as BB or below. Panel C shows the overall RMSE by firm size, with small firms defined as those below the median market capitalization and large firms as those at or above the median.



(a) Pooled RMSE



(b) Pooled RMSE by Firm Credit Rating



(c) Pooled RMSE by Firm Market Capitalization

Figure 5. Time Series of Out-of-Sample Prediction Error. This figure plots the monthly out-of-sample root mean squared error (RMSE) for all machine-learning algorithms over the period from March 2015 to December 2020.

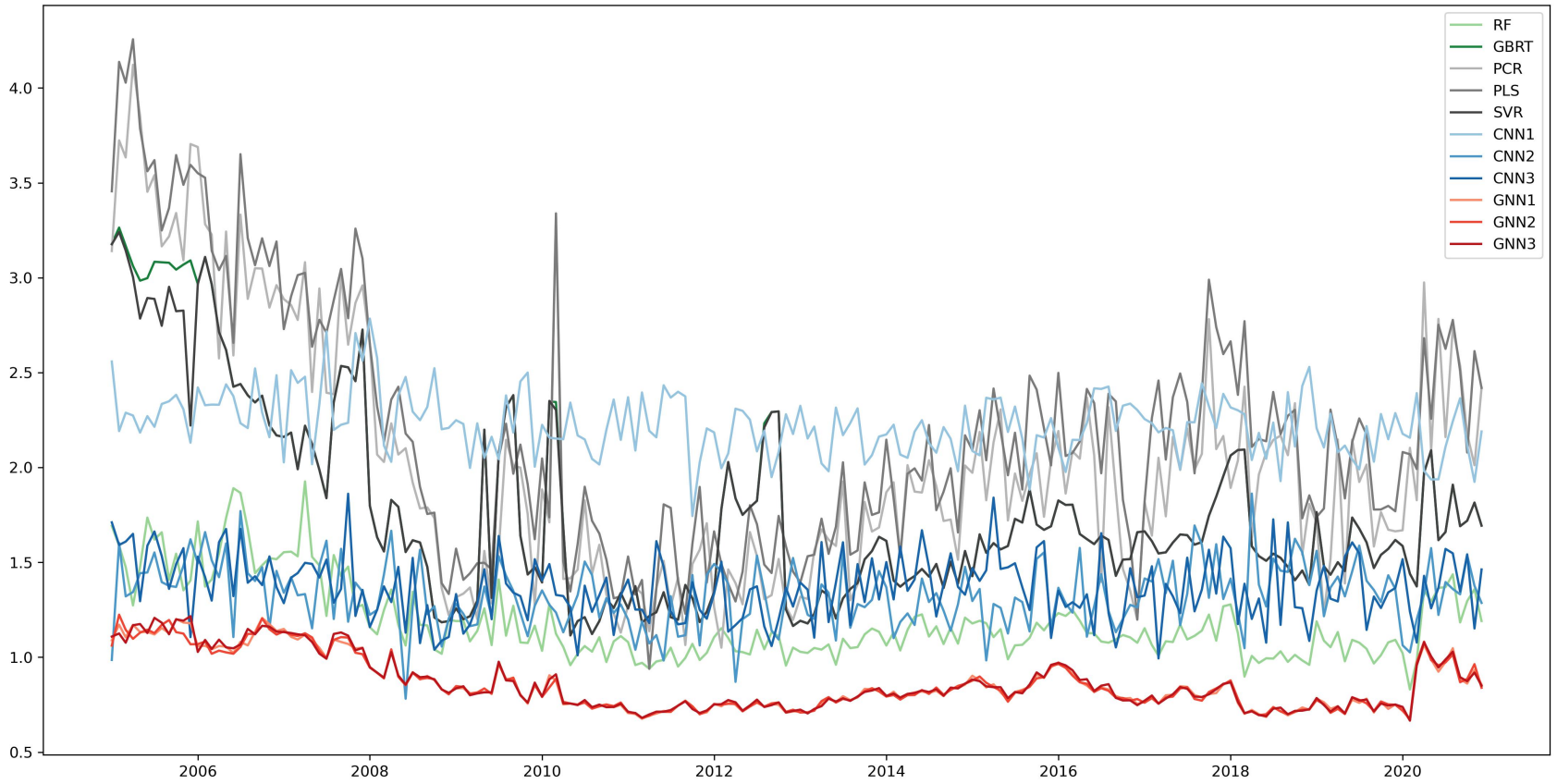
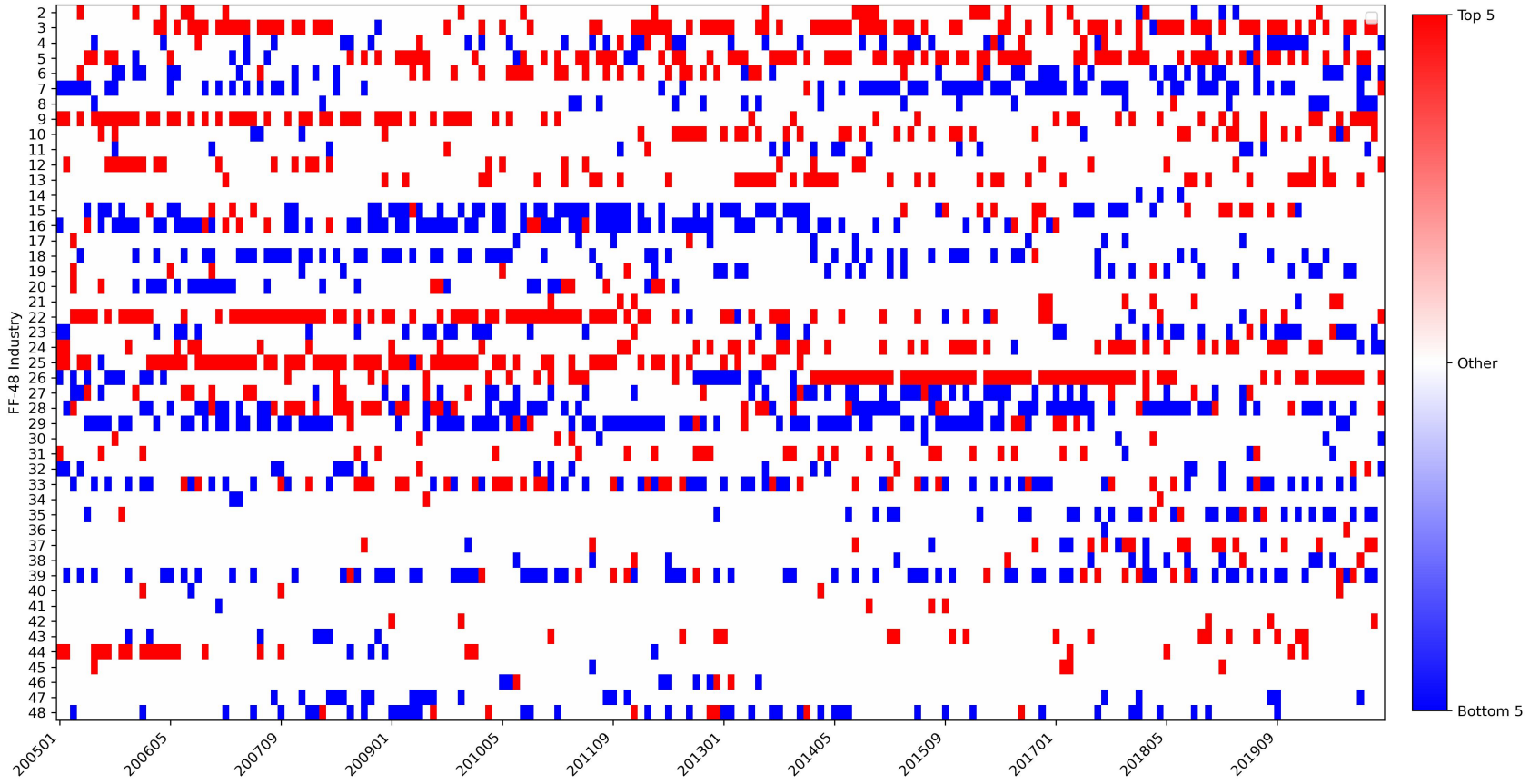


Figure 6. Time-Varying Network-Attributable Spread Changes Across Industries. This figure presents a heatmap of the time-varying network-attributable spread change (NSC) across industries. For each month and each Fama–French 48 industry, we compute NSC, defined as the average absolute difference between the predicted log CDS spread from GNN2 and that from CNN2, normalized by the average log CDS spread within the industry. At each month, industries are ranked by NSC. The five industries with the largest network-attributable spread change (NSC) are highlighted in red, while the five industries with the smallest network-attributable spread change are highlighted in blue; all remaining industries are shown in white.



References

Acemoglu, Daron, Ufuk Akcigit, and William Kerr, 2016, Networks and the macroeconomy: An empirical exploration, *Nber Macroeconomics Annual* 30, 273–335.

Acemoglu, Daron, Vasco M Carvalho, Asuman Ozdaglar, and Alireza Tahbaz-Salehi, 2012, The network origins of aggregate fluctuations, *Econometrica* 80, 1977–2016.

Acemoglu, Daron, Asuman Ozdaglar, and Alireza Tahbaz-Salehi, 2017, Microeconomic origins of macroeconomic tail risks, *American Economic Review* 107, 54–108.

Aït-Sahalia, Yacine, Julio Cacho-Diaz, and Roger Laeven, 2015, Modeling financial contagion using mutually exciting jump processes, *Journal of Financial Economics* 117, 585–606.

Almeida, Heitor, and Thomas Philippon, 2007, The risk-adjusted cost of financial distress, *The Journal of Finance* 62, 2557–2586.

Altman, Edward I, 1968, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *The Journal of Finance* 23, 589–609.

Ang, Andrew, and Francis A. Longstaff, 2013, Systemic sovereign credit risk: lessons from the u.s. and europe, *Journal of Monetary Economics* 60, 493–510.

Azizpour, Shahriar, Kay Giesecke, and Gustavo Schwenkler, 2018, Exploring the sources of default clustering, *Journal of Financial Economics* 129, 154–183.

Bao, Jack, Kewei Hou, and Shaojun Zhang, 2023, Systematic default and return predictability in the stock and bond markets, *Journal of Financial Economics* 149, 349–377.

Bao, Jack, and Jun Pan, 2013, Relating equity and credit markets through structural models: evidence from volatilities, *Journal of Finance* 68, 2359–2398.

Benzoni, Luca, Pierre Collin-Dufresne, Robert S Goldstein, and Jean Helwege, 2015, Modeling credit contagion via the updating of fragile beliefs, *The Review of Financial Studies* 28, 1960–2008.

Berndt, Antje, Rohan Douglas, Darrell Duffie, and Mark Ferguson, 2018, Corporate credit risk premia, *Review of Finance* 22, 419–454.

Bharath, Sreedhar T, and Tyler Shumway, 2008, Forecasting default with the merton distance to default model, *The Review of Financial Studies* 21, 1339–1369.

Blasques, Francisco, Siem Jan Koopman, Andre Lucas, and Julia Schaumburg, 2016, Spillover dynamics for systemic risk measurement using spatial financial time series models, *Journal of Econometrics* 195, 211–223.

Boyarchenko, Nina, and Or Shachar, 2020, The evolving market for u.s. sovereign credit risk, *Liberty Street Economics* .

Breiman, Leo, 2001, Random forests, *Machine learning* 45, 5–32.

Campbell, John Y, Jens Hilscher, and Jan Szilagyi, 2008, In search of distress risk, *The Journal of Finance* 63, 2899–2939.

Capponi, Agostino, J Antonio Sidaoui, and Jiacheng Zou, 2025, Graph machine learning for asset pricing: Traversing the supply chain, *Available at SSRN 5031617* .

Carvalho, Vasco, and Xavier Gabaix, 2013, The great diversification and its undoing, *American Economic Review* 103, 1697–1727.

Carvalho, Vasco M, 2008, *Aggregate fluctuations and the network structure of intersectoral trade* (The University of Chicago).

Chen, Belinda, 2023, Network factors for idiosyncratic volatility spillover, *Available at SSRN 4579385* .

Chen, Cathy Yi-Hsuan, Wolfgang Karl Härdle, and Yarema Okhrin, 2019, Tail event driven networks of sifis, *Journal of Econometrics* 208, 282–298.

Collin-Dufresne, Pierre, Robert S. Goldstein, and J. Spencer Martin, 2001, The determinants of credit spread changes, *Journal of Finance* 56, 2177–2207.

Das, Sanjiv R., Darrell Duffie, Nikunj Kapadia, and Leandro Saita, 2007, Common failings: how corporate defaults are correlated, *Journal of Finance* 62, 93–117.

Demirer, Mert, Francisdasdu X Diebold, Laura Liu, and Kamil Yilmaz, 2018, Estimating global bank network connectedness, *Journal of Applied Econometrics* 33, 1–15.

Dew-Becker, Ian, 2023, Tail risk in production networks, *Econometrica* 91, 2089–2123.

Diebold, Francis X, and Kamil Yılmaz, 2014, On the network topology of variance decompositions: Measuring the connectedness of financial firms, *Journal of Econometrics* 182, 119–134.

Duffie, Darrell, 1999, Credit swap valuation, *Financial Analysts Journal* 55, 73–87.

Duffie, Darrell, and Nicolae Gârleanu, 2001, Risk and valuation of collateralized debt obligations, *Financial Analysts Journal* 57, 41–59.

Duffie, Darrell, and David Lando, 2001, Term structures of credit spreads with incomplete accounting information, *Econometrica* 69, 633–664.

Duffie, Darrell, and Jun Liu, 2001, Floating–fixed credit spreads, *Financial Analysts Journal* 57, 76–87.

Duffie, Darrell, and Jun Pan, 1997, An overview of value at risk, *Journal of Derivatives* 4, 7–49.

Duffie, Darrell, Jun Pan, and Kenneth J. Singleton, 2000, Transform analysis and asset pricing for affine jump-diffusions, *Econometrica* 68, 1343–1376.

Duffie, Darrell, Lasse Heje Pedersen, and Kenneth J. Singleton, 2003, Modeling sovereign yield spreads: a case study of russian debt, *Journal of Finance* 58, 119–159.

Duffie, Darrell, Leandro Saita, and Ke Wang, 2007, Multi-period corporate default prediction with stochastic covariates, *Journal of Financial Economics* 83, 635–665.

Duffie, Darrell, and Kenneth J. Singleton, 1999, Modeling term structures of defaultable bonds, *Review of Financial Studies* 12, 687–720.

Engle, Robert, and Bryan Kelly, 2012, Dynamic equicorrelation, *Journal of Business & Economic Statistics* 30, 212–228.

Fama, Eugene F, and Kenneth R French, 2016, Dissecting anomalies with a five-factor model, *The Review of Financial Studies* 29, 69–103.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, 2020, Dissecting characteristics nonparametrically, *The Review of Financial Studies* 33, 2326–2377.

Gabaix, Xavier, 2011, The granular origins of aggregate fluctuations, *Econometrica* 79, 733–772.

Galil, Koresh, Offer Moshe Shapir, Dan Amiram, and Uri Ben-Zion, 2014, The determinants of cds spreads, *Journal of Banking & Finance* 41, 271–282.

Getmansky, Mila, Giulio Girardi, and Craig Lewis, 2016, Interconnectedness in the cds market, *Financial Analysts Journal* 72, 62–82.

Giesecke, Kay, 2002, *An exponential model for dependent defaults* (Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät).

Giesecke, Kay, 2004, Correlated default with incomplete information, *Journal of Banking & Finance* 28, 1521–1545.

Giesecke, Kay, and Stefan Weber, 2004, Cyclical correlations, credit contagion, and portfolio losses, *Journal of Banking & Finance* 28, 3009–3036.

Gouriéroux, Christian, Alain Monfort, and Jean-Paul Renne, 2014, Pricing default events: Surprise, exogeneity and contagion, *Journal of Econometrics* 182, 397–411.

Green, Jeremiah, John RM Hand, and X Frank Zhang, 2017, The characteristics that provide independent information about average us monthly stock returns, *The Review of Financial Studies* 30, 4389–4436.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020, Empirical asset pricing via machine learning, *The Review of Financial Studies* 33, 2223–2273.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2021, Autoencoder asset pricing models, *Journal of Econometrics* 222, 429–450.

Härdle, Wolfgang Karl, Weining Wang, and Lining Yu, 2016, Tenet: Tail-event driven network risk, *Journal of Econometrics* 192, 499–513.

Hawkes, Alan G, 1971, Spectra of some self-exciting and mutually exciting point processes, *Biometrika* 58, 83–90.

Herskovic, Bernard, 2018, Networks in production: Asset pricing implications, *The Journal of Finance* 73, 1785–1818.

Herskovic, Bernard, Bryan Kelly, Hanno Lustig, and Stijn Van Nieuwerburgh, 2016, The common factor in idiosyncratic volatility: Quantitative asset pricing implications, *Journal of Financial Economics* 119, 249–283.

Herskovic, Bernard, Bryan Kelly, Hanno Lustig, and Stijn Van Nieuwerburgh, 2020, Firm volatility in granular networks, *Journal of Political Economy* 128, 4097–4162.

Hou, Kewei, Chen Xue, and Lu Zhang, 2020, Replicating anomalies, *The Review of financial studies* 33, 2019–2133.

Hu, Grace Xing, Jun Pan, and Jiang Wang, 2013, Noise as information for illiquidity, *Journal of Finance* 68, 2341–2382.

Ioffe, Sergey, and Christian Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *International Conference on Machine Learning*, 448–456, PMLR.

Jacobson, Tor, and Erik Von Schedvin, 2015, Trade credit and the propagation of corporate failure: An empirical analysis, *Econometrica* 83, 1315–1371.

Jarrow, Robert A, and Fan Yu, 2001, Counterparty risk and the pricing of defaultable securities, *The Journal of Finance* 56, 1765–1799.

Jorion, Philippe, and Gaiyan Zhang, 2007, Good and bad credit contagion: evidence from credit default swaps, *Journal of Financial Economics* 84, 860–883.

Jorion, Philippe, and Gaiyan Zhang, 2009, Credit contagion from counterparty risk, *Journal of Finance* 64, 2053–2087.

Kelly, Bryan, and Hao Jiang, 2014, Tail risk and asset prices, *The Review of Financial Studies* 27, 2841–2871.

Kelly, Bryan, Semyon Malamud, and Kangying Zhou, 2024, The virtue of complexity in return prediction, *The Journal of Finance* 79, 459–503.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, 2019, Characteristics are covariances: A unified model of risk and return, *Journal of Financial Economics* 134, 501–524.

Kipf, Thomas N, and Max Welling, 2016, Semi-supervised classification with graph convolutional networks, *arXiv preprint arXiv:1609.02907* .

Kitwiwattanachai, Chanatip, 2015, Learning network structure of financial institutions from cds data, *Available at SSRN 2533606* .

Kitwiwattanachai, Chanatip, and Neil D. Pearson, 2015, Inferring Correlations of Asset Values and Distances-to-Default from CDS Spreads: A Structural Model Approach, *The Review of Asset Pricing Studies* 5, 112–154.

Kryzanowski, Lawrence, Stylianos Perrakis, and Rui Zhong, 2017, Price discovery in equity and cds markets, *Journal of Financial Markets* 35, 21–46.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, 2015, Deep learning, *nature* 521, 436–444.

Liu, Lily Y, 2022, Estimating loss given default from cds under weak identification, *Journal of Financial Econometrics* 20, 310–344.

Longstaff, Francis A., Jun Pan, Lasse H. Pedersen, and Kenneth J. Singleton, 2011, How sovereign is sovereign credit risk, *American Economic Journal: Macroeconomics* 3, 75–103.

Merton, Robert C, 1974, On the pricing of corporate debt: The risk structure of interest rates, *The Journal of finance* 29, 449–470.

Monfort, Alain, Fulvio Pegoraro, Jean-Paul Renne, and Guillaume Roussellet, 2021, Affine modeling of credit risk, pricing of credit events, and contagion, *Management Science* 67, 3674–3693.

Thekumparampil, Kiran K, Chong Wang, Sewoong Oh, and Li-Jia Li, 2018, Attention-based graph neural network for semi-supervised learning, *arXiv preprint arXiv:1803.03735* .

Uddin, Ajim, Xinyuan Tao, and Dantong Yu, 2021, Attention based dynamic graph learning framework for asset pricing, in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1844–1853.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017, Attention is all you need, *Advances in neural information processing systems* 30.

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, 2017, Graph attention networks, *arXiv preprint arXiv:1710.10903* .

Wang, Daixin, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi, 2019, A semi-supervised graph attentive network for financial fraud detection, in *2019 IEEE International Conference on Data Mining (ICDM)*, 598–607, IEEE.

Welch, Ivo, and Amit Goyal, 2008, A comprehensive look at the empirical performance of equity premium prediction, *The Review of Financial Studies* 21, 1455–1508.

Zhang, Chao, Xingyue Pu, Mihai Cucuringu, and Xiaowen Dong, 2023, Graph neural networks for forecasting multivariate realized volatility with spillover effects, *arXiv preprint arXiv:2308.01419* .

Internet Appendix

A CES Production Network Foundations and Credit-Risk Spillovers

This appendix derives the structural objects used in the main text. Section A.1 derives the CES price recursion and the high-order exposure matrix using the firm-level notation of the main text. Section A.2 reports the fully substituted nonlinear CDS-spread expression. Sections A.3, A.4, and A.5 prove the three model implications.

A.1 CES Production and Unit Costs

Firm i produces output according to

$$Y_{i,t} = \xi_t Z_{i,t} L_{i,t}^{1-\alpha_i} \left(\sum_{j=1}^{N} \omega_{ij,t}^{1/\sigma_i} X_{ij,t}^{(\sigma_i-1)/\sigma_i} \right)^{\alpha_i \sigma_i/(\sigma_i-1)}.$$

The CES input price index associated with firm i's intermediate-input bundle is

$$C_{M,i,t} = \left(\sum_{j=1}^{N} \omega_{ij,t} P_{j,t}^{1-\sigma_i} \right)^{1/(1-\sigma_i)}.$$

This price index is the unit cost of one unit of the CES intermediate aggregate. With inelastic labor normalized to $L_{i,t} = 1$, the wage normalized to one, and constants that do not affect relative propagation suppressed, perfect competition implies

$$P_{i,t} = \xi_t^{-1} Z_{i,t}^{-1} C_{M,i,t}^{\alpha_i}.$$

Taking logs and removing the aggregate component gives

$$p_{i,t} = -z_{i,t} + \frac{\alpha_i}{1-\sigma_i} \log \left(\sum_{j=1}^{N} \omega_{ij,t} \exp\{(1-\sigma_i) p_{j,t}\} \right),$$

which is the relative-price recursion used in the main text.

The CES expenditure share on input j is

$$s_{ij,t} = \frac{\omega_{ij,t} \exp\{(1-\sigma_i) p_{j,t}\}}{\sum_{k=1}^{N} \omega_{ik,t} \exp\{(1-\sigma_i) p_{k,t}\}}.$$

The first-order differential of the price recursion is

$$dp_{i,t} = -dz_{i,t} + \alpha_i \sum_{j=1}^{N} s_{ij,t} dp_{j,t}.$$

Stacking this expression across firms yields

$$dp_t = -dz_t + B_t dp_t, \qquad B_{ij,t} = \alpha_i s_{ij,t}.$$

If $\rho(B_t) < 1$, then

$$dp_t = -\mathcal{L}_t dz_t, \qquad \mathcal{L}_t = (I - B_t)^{-1} = \sum_{m=0}^{\infty} B_t^m. \tag{21}$$

The element $\ell_{ij,t} \equiv \mathcal{L}_{ij,t}$ is the local high-order level exposure from supplier firm j to customer firm i.

The multiplier $\mathcal{L}_t$ is distinct from the standard Leontief inverse. A standard production-network inverse is typically constructed from primitive cost shares $C_{ij,t} = \alpha_i \omega_{ij,t}$ and used to solve for sales shares, Domar weights, or gross-output responses, depending on the production environment. The multiplier in Equation (21) is instead constructed from the endogenous expenditure shares $s_{ij,t}$. It is therefore a local first-order and higher-order shock-pass-through multiplier of the equilibrium CES expenditure-share network. In the Cobb–Douglas limit, $s_{ij,t} = \omega_{ij,t}$, so the algebraic form coincides with the standard inverse. Outside that limit, equilibrium prices and substitution elasticities change the propagation weights.

Under conditional orthogonality of idiosyncratic shocks, the raw variance contribution from supplier j to customer i is

$$v_{ij,t} = \ell_{ij,t}^2 \sigma_{u,j,t}^2.$$

The normalized idiosyncratic-volatility spillover share is

$$a_{ij,t} = \frac{\ell_{ij,t}^2 \sigma_{u,j,t}^2}{\sum_{k=1}^{N} \ell_{ik,t}^2 \sigma_{u,k,t}^2}.$$

This normalized share is a second-moment object. It is distinct from $\ell_{ij,t}$, which governs cash-flow levels.

A.2 Fully Substituted Credit-Spread Function

This subsection reports the nonlinear mapping from the production network to CDS spreads after substituting the network cash-flow and volatility channels into the Merton distance-to-default approximation. Define

$$X_{i,t}^L = \sum_{j=1}^{N} \ell_{ij,t} u_{j,t}, \tag{22}$$

$$X_{i,t}^V = \sum_{j=1}^{N} \ell_{ij,t}^2 \sigma_{u,j,t}^2. \tag{23}$$

Let

$$\widetilde{E}_{i,t} \equiv \bar{E}_{i,t} - \kappa_{i,t}^E X_{i,t}^L = \bar{E}_{i,t} - \kappa_{i,t}^E \sum_{j=1}^{N} \ell_{ij,t} u_{j,t}. \tag{24}$$

The Bharath–Shumway asset-value approximation gives

$$V_{i,t} = \widetilde{E}_{i,t} + F_i.$$

The equity-volatility channel is

$$\sigma_{E,i,t} = \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} X^V_{i,t}\right)^{1/2} = \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}\right)^{1/2}.$$

The asset-volatility approximation gives

$$\sigma_{V,i,t} = \frac{\widetilde{E}_{i,t}}{\widetilde{E}_{i,t} + F_i} \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}\right)^{1/2}.$$

Let $\tau \equiv T - t$. Substituting these expressions into the Merton distance-to-default formula gives

$$DD_{i,t} = \frac{\log\left[\frac{\widetilde{E}_{i,t}+F_i}{F_i}\right] + \left[r - \frac{1}{2}\left(\frac{\widetilde{E}_{i,t}}{\widetilde{E}_{i,t}+F_i}\right)^2 \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}\right)\right] \tau}{\left(\frac{\widetilde{E}_{i,t}}{\widetilde{E}_{i,t}+F_i}\right) \left(\bar{\sigma}^2_{E,i,t} + \kappa^{\sigma}_{i,t} \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}\right)^{1/2} \sqrt{\tau}}. \tag{25}$$

After replacing $\widetilde{E}_{i,t}$ with Equation (24), this expression shows explicitly that distance to default is a nonlinear function of the level-channel terms $\{\ell_{ij,t} u_{j,t}\}_{j=1}^{N}$ and the volatility-channel terms $\{\ell^2_{ij,t} \sigma^2_{u,j,t}\}_{j=1}^{N}$.

The risk-neutral default probability is

$$PD^{\mathbb{Q}}_{i,t} = \Phi(-DD_{i,t}),$$

and the zero-coupon CDS approximation gives

$$s^{CDS}_{i,t} = \frac{1 - R}{\tau} \Phi(-DD_{i,t}). \tag{26}$$

Together, Equations (25)–(26) provide the fully substituted nonlinear mapping from the input-output network to CDS spreads.

A.3 Proof of Implication 1

The proof follows by substituting the two network exposures into the Merton distance-to-default mapping. From Equations (22) and (23), the cash-flow level exposure of firm i is

$$X^L_{i,t} = \sum_{j=1}^{N} \ell_{ij,t} u_{j,t},$$

and, under conditional orthogonality of idiosyncratic shocks, the cash-flow volatility exposure is

$$X^V_{i,t} = \sum_{j=1}^{N} \ell^2_{ij,t} \sigma^2_{u,j,t}.$$

The equity mapping in the main text gives

$$E_{i,t} = \bar{E}_{i,t} - \kappa^E_{i,t} X^L_{i,t}, \qquad \sigma_{E,i,t} = \left(\bar{\sigma}^2_{E,i,t} + \kappa^\sigma_{i,t} X^V_{i,t}\right)^{1/2}.$$

The Bharath–Shumway approximations then map $E_{i,t}$ and $\sigma_{E,i,t}$ into $V_{i,t}$ and $\sigma_{V,i,t}$, and the Merton formula maps $V_{i,t}$ and $\sigma_{V,i,t}$ into $DD_{i,t}$. Therefore, $DD_{i,t}$ is a nonlinear function of the full network multiplier $\mathcal{L}_t = [\ell_{ij,t}]^N_{i,j=1}$ through both $X^L_{i,t}$ and $X^V_{i,t}$. Substitution into the zero-coupon CDS expression gives Equation (10).

Because $\mathcal{L}_t = (I - B_t)^{-1} = \sum_{m=0}^{\infty} B_t^m$, the entries $\ell_{ij,t}$ incorporate both direct links and higher-order input-output paths. Conditional orthogonality removes covariance terms from the volatility channel, so each supplier's second-moment contribution is $\ell^2_{ij,t}\sigma^2_{u,j,t}$. Hence, orthogonal negative idiosyncratic productivity shocks to all firms propagate through both direct and indirect input-output relationships and enter the cross section of CDS spreads through the level and volatility channels. This proves Implication 1.

A.4 Proof of Implication 2

We first show how the central-supplier topology amplifies a finite shock. Let supplier c satisfy Assumption 1. For any customer firm $i \in \mathcal{I}$, the local direct exposure from supplier c to customer i is

$$D_{ic,t} \equiv B_{ic,t} = \alpha_i s_{ic,t}.$$

The high-order exposure is

$$\ell_{ic,t} = \mathcal{L}_{ic,t} = \sum_{m=0}^{\infty} (B_t^m)_{ic}.$$

Because B_t has nonnegative elements, every term in the series is nonnegative. Consider the subset of paths in which the shock remains in supplier c's self-loop for m rounds and then passes from supplier c to customer i through the direct link. These paths contribute $D_{ic,t} B^m_{cc,t}$. Therefore,

$$\ell_{ic,t} \geq D_{ic,t} \sum_{m=0}^{\infty} B^m_{cc,t} = \frac{D_{ic,t}}{1 - B_{cc,t}}, \tag{27}$$

provided $B_{cc,t} < 1$. Assumption 1 gives $s_{ic,t} \geq 1 - \varepsilon$ and $s_{cc,t} \geq 1 - \varepsilon$, so

$$D_{ic,t} = \alpha_i s_{ic,t} \geq \alpha_i (1 - \varepsilon), \qquad B_{cc,t} = \alpha_c s_{cc,t} \geq \alpha_c (1 - \varepsilon). \tag{28}$$

Combining Equations (27) and (28) gives

$$\ell_{ic,t} \geq \frac{\alpha_i (1 - \varepsilon)}{1 - \alpha_c (1 - \varepsilon)}.$$

Thus, as $\alpha_c(1 - \varepsilon) \uparrow 1$, the lower bound on $\ell_{ic,t}$ diverges.

For a finite supplier shock $u_{c,t} > 0$, the customer firm's cash-flow level exposure includes the term $\ell_{ic,t} u_{c,t}$, and its volatility exposure includes the term $\ell^2_{ic,t}\sigma^2_{u,c,t}$. The fully substituted mapping in Ap-

pendix A.2 then implies that sufficiently large $\ell_{ic,t}$ lowers the customer's distance to default through both the equity-value channel and the equity-volatility channel. Therefore, for any target probability $q \in (0,1)$, there exists a degree of centrality such that the customer firm's risk-neutral default probability is at least q. The supplier remains solvent by the maintained condition $DD_{c,t}(u_{c,t}) > \bar{d}_c$. In the limiting bottleneck case $\alpha_c(1-\varepsilon) \uparrow 1$, the customer firm's default probability converges to one, which proves Equations (13) and (14).

The feedback discussion in the main text follows from the same hub-and-spoke structure. If peripheral firms default, the goods they provide to supplier c become unavailable or more costly. Even if these peripheral input shares are thin, they can feed back into supplier c's input-cost aggregator. With complementary inputs, the loss of outside inputs can prevent completion of production. With substitutable inputs, the supplier reallocates toward its own output, reinforcing the self-input loop $B_{cc,t} = \alpha_c s_{cc,t}$. Thus, the topology can amplify the initial finite shock and can feed customer distress back to the central supplier.

A.5 Proof of Implication 3

Consider a shock $z_t = -xe_c$ with $x \to \infty$. We look for price asymptotics of the form

$$p_{i,t}(-xe_c) = \mu_i^c + \phi_i^c x + o(x).$$

Substituting this expression into the CES price recursion, the productivity term contributes x if $i = c$ and zero otherwise. The CES log-sum term is

$$\frac{\alpha_i}{1-\sigma_i} \log\left(\sum_{j \in \mathcal{S}_i} \omega_{ij} \exp\{(1-\sigma_i)(\mu_j^c + \phi_j^c x + o(x))\} \right).$$

If $\sigma_i < 1$, then $1 - \sigma_i > 0$, and the largest ϕ_j^c determines the leading term of the log-sum as $x \to \infty$. If $\sigma_i > 1$, then $1 - \sigma_i < 0$; after division by the negative number $1 - \sigma_i$, the smallest ϕ_j^c determines the leading term. If $\sigma_i = 1$, the Cobb–Douglas limit gives the weighted average. Matching coefficients on x gives

$$\phi_i^c = \mathbf{1}\{i = c\} + \alpha_i \begin{cases} \max_{j \in \mathcal{S}_i} \phi_j^c, & \sigma_i < 1, \\ \sum_{j \in \mathcal{S}_i} \omega_{ij} \phi_j^c, & \sigma_i = 1, \\ \min_{j \in \mathcal{S}_i} \phi_j^c, & \sigma_i > 1. \end{cases}$$

This is the tail-recursion expression stated in Equation (15).

If $\phi_i^c > 0$, the cost and price of firm i's output grow linearly with the shock size x. Along the affected path, the implied cash-flow loss exposure and volatility exposure diverge. The fully substituted CDS-spread mapping in Appendix A.2 then implies that distance to default tends to $-\infty$ and the risk-neutral default probability tends to one. This proves Equation (16).

B CDS Distribution Across Sectors

Table 3. Firm Distribution Across Sectors
This table presents the distribution of firms in our sample across sectors, categorized by the first two digits of their NAICS code. It reports the number of firms and the average log CDS spread for each sector.

Sector Name	First 2 Digit of NAICS	Firm Count	Mean log CDS
Accommodation and Food Services	72	16	-4.79
Administrative and Support and Waste Management and Remediation Services	56	11	-5.09
Construction	23	14	-4.10
Finance and Insurance	52	117	-4.73
Health Care and Social Assistance	62	16	-4.20
Information	51	54	-4.42
Manufacturing	31, 32, 33	268	-4.83
Mining, Quarrying, and Oil and Gas Extraction	21	48	-4.51
Professional, Scientific, and Technical Services	54	26	-4.56
Real Estate and Rental and Leasing	53	33	-4.33
Retail Trade	44, 45	45	-4.58
Transportation and Warehousing	48, 49	31	-4.70
Utilities	22	55	-4.88
Wholesale Trade	42	38	-4.68

C Firm Level Characteristics

Table 4. Firm-Level Characteristics
This table presents 94 firm-level characteristics constructed following Gu, Kelly, and Xiu.

Acronym	Firm characteristic	Acronym	Firm characteristic
absacc	Absolute accruals	*divo*	Dividend omission
acc	Working capital accruals	*dolvol*	Dollar trading volume
aeavol	Abnormal earnings announcement volume	*dy*	Dividend to price
age	# years since first Compustat coverage	*ear*	Earnings announcement return
agr	Asset growth	*egr*	Growth in common shareholder equity
baspread	Bid-ask spread	*ep*	Earnings to price
beta	Beta	*gma*	Gross profitability
betasq	Beta squared	*grcapx*	Growth in capital expenditures
bm	Book-to-market	*grltnoa*	Growth in long term net operating assets
bm_ia	Industry-adjusted book to market	*herf*	Industry sales concentration
cash	Cash holdings	*hire*	Employee growth rate
cashdebt	Cash flow to debt	*idiovol*	Idiosyncratic return volatility
cashpr	Cash productivity	*ill*	Illiquidity
cfp	Cash flow to price ratio	*indmom*	Industry momentum
cfp_ia	Industry-adjusted cash flow to price ratio	*invest*	Capital expenditures and inventory
chatoia	Industry-adjusted change in asset turnover	*lev*	Leverage
chcsho	Change in shares outstanding	*lgr*	Growth in long-term debt
chempia	Industry-adjusted change in employees	*maxret*	Maximum daily return
chinv	Change in inventory	*mom12m*	12-month momentum
chmom	Change in 6-month momentum	*mom1m*	1-month momentum
chpmia	Industry-adjusted change in profit margin	*mom36m*	36-month momentum
chtx	Change in tax expense	*mom6m*	6-month momentum

Table 5. Firm-Level Characteristics (Continued)

cinvest	Corporate investment	*ms*	Financial statement score
convind	Convertible debt indicator	mvel1	log market equity
currat	Current ratio	*mve_ia*	Industry-adjusted size
depr	Depreciation / PP&E	*nincr*	Number of earnings increases
divi	Dividend initiation	*operprof*	Operating profitability
orgcap	Organizational capital	*roeq*	Return on equity
pchcapx_ia	Industry adjusted % change in capital expenditures	*roic*	Return on invested capital
pchcurrat	% change in current ratio	*rsup*	Revenue surprise
pchdepr	% change in depreciation	*salecash*	Sales to cash
pchgm_pchsale	% change in gross margin - % change in sales	*saleinv*	Sales to inventory
pchquick	% change in quick ratio	*salerec*	Sales to receivables
pch-sale_pchinvt	% change in sales - % change in inventory	*secured*	Secured debt
pch-sale_pchrect	% change in sales - % change in A/R	*securedind*	Secured debt indicator
pch-sale_pchxsga	% change in sales - % change in SG&A	*sgr*	Sales growth
pchsaleinv	% change sales-to-inventory	*sin*	Sin stocks
pctacc	Percent accruals	*SP*	Sales to price
pricedelay	Price delay	*std_dolvol*	Volatility of liquidity (dollar trading volume)
ps	Financial statements score	*std_turn*	Volatility of liquidity (share turnover)
quick	Quick ratio	*stdacc*	Accrual volatility
rd	R&D increase	*stdcf*	Cash flow volatility
rd_mve	R&D to market capitalization	*tang*	Debt capacity/firm tangibility
rd_sale	R&D to sales	*tb*	Tax income to book income
realestate	Real estate holdings	*turn*	Share turnover
retvol	Return volatility	*roavol*	Earnings volatility
roaq	Return on assets	*zerotrade*	Zero trading days

D Algorithm Details

D.1 Graph Neural Networks

The Graph Neural Network (GNN) algorithm encompasses two primary frameworks: the updating scheme, which relates to intra-layer design, and the message passing scheme, which is inter-layer design. Figure 1 illustrates these schemes, with panel A showing the intra-layer scheme and panel B depicting the inter-layer scheme.

The updating scheme is articulated as follows:

FIGURE 1 ABOUT HERE

$$h_i^{(k+1)} = AGG\left\{ACT\left(DP\left(BN\left(\omega^{(k)}\hat{h}_i^{(k)} + b^{(k)}\right)\right)\right), j \in \mathcal{N}(i)\right\}, \tag{29}$$

where $h_i^{(k+1)}$ is the node embedding for layer $k+1$, $AGG(\cdot)$ is the aggregation function, $ACT(\cdot)$ is the nonlinear activation function, $DP(\cdot)$ is the dropout function, $BN(\cdot)$ is the batch normalization function, $\omega^{(k)}$ is the trainable weight matrix, $b^{(k)}$ is the bias term, $\hat{h}_i^{(k)}$ is the normalized embedding of node i at layer k, and $\mathcal{N}(i)$ denotes the local neighborhood of node i.

Equation (29) states the rule for updating node embeddings within a GNN comprising a single hidden layer. The embeddings from neighboring nodes are aggregated, and the resulting vector undergoes a series of transformations: batch normalization, dropout, and nonlinear activation, followed by the application of a trainable weight matrix and a bias term. We use batch normalization (see (Ioffe and Szegedy, 2015)) to help stabilize and accelerate the training of deep neural networks. We employ dropout to prevent overfitting and use the rectified linear unit (ReLU) as an activation function to introduce nonlinearity. Finally, we aggregate node characteristics by using edge-weighted average node characteristics to reflect the relative importance of neighboring information.

To capture the dynamic network effect, represented by Panel A in Figure 1, we incorporate an additional technique. For each hidden layer, we adopt Long Short-Term Memory (LSTM) to capture temporal graph dependencies. It's important to note that we do not apply LSTM to each input variable; instead, we use it to aggregate the temporal information from the GCN neuron outputs. As the basic GCN framework takes the graph as fixed, adding LSTM between neurons introduces temporal dependencies. To avoid model overcomplexity and misspecification, we fine-tune all parameters for both GCN and LSTM, which is elaborated in Section 3.2.

The message passing scheme is described as follows. Each $\hat{h}_i^{(k)}$ in a minibatch covers both the node i and its associated information, as delineated by the subsequent equation:

$$\hat{h}_i^{(k)} = h_i^{(k)} \bigoplus_{j \in \mathcal{N}(i)} (a_{i,j}), \tag{30}$$

where $\hat{h}_i^{(k)}$ is the normalized embedding of node i at layer k, $h_i^{(k)}$ is the embedding of node i at layer k, $\bigoplus$ represents the concatenation operation and $\mathcal{N}(i)$ is the neighborhood of node i.

In equation (30) , $a_{i,j}$ denotes the edge characteristic (i.e., the directed contagion effect) from node j to node i. The matrix collecting all elements $a_{i,j}$ is defined as A, i.e., $A \equiv [a_{ij}]$. A will also be referred to as the adjacency matrix.

To encode the network effect, we follow the conventional graph convolutional network (GCN) algorithm proposed by Kipf and Welling (2016). Under GCN, the network information is encoded as follows:

$$\hat{H}^{(k)} = \begin{cases} D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} H^{(k)} W^{(k)} & \text{if } A \text{ is symmetric} \\ D^{-1} \hat{A} H^{(k)} W^{(k)} & \text{if } A \text{ is asymmetric} \end{cases} \tag{31}$$

where $\hat{H}^{(k)}$ is the output of the k-th layer, $\hat{A} = A + I$ with I being the identity matrix, D is the out-degree matrix, $H^{(k)}$ is the input to the k-th layer, and $W^{(k)}$ is the trainable weight matrix for the k-th layer. In equation (31), the out-degree matrix D is a diagonal matrix where the diagonal elements are equal to the column sums of A.

Finally, to address the challenge of an unbalanced panel where each CDS only exists for a certain period throughout the time, we apply a masking scheme to the GNN algorithm. We mask the nodes (i.e., firms) and the corresponding edges if they do not exist.

D.2 GNN-Attention

To elucidate the mechanism within the neuron training, we enhance the basic GNN algorithm with a technique called 'attention'[14].We refer to this enhanced algorithm as GNN-attention. While a basic GNN is often a black box with nonlinearity and complexity, incorporating attention layers allows us to assign different weights to nodes and edges to minimize the training error. These weights, when output, can effectively reveal which nodes or edges are given more 'attention'.

We incorporate node attention into the intra-layer design. We initiate a trainable $n \times 1$ node attention vector, which is updated from layer to layer. This vector is then aggregated with the node features matrix to dynamically weight neighboring nodes based on their importance scores, enabling the GNN to concentrate on more relevant information during the intra-layer update. The algorithm is as follows:

$$\begin{cases} \text{node attention}^{(k+1)} = DP\left(ACT\left(BN\left(\text{node attention}^{(k)}\right)\right)\right) \\ h_i^{(k+1)} = AGG_2\left\{AGG_1\left\{DP\left(ACT\left(BN\left(\omega^{(k)}\hat{h}_i^{(k)} + b^{(k)}\right)\right)\right), \text{Node attention}^{(k)}\right\}, j \in \mathcal{N}(i)\right\} \end{cases} \tag{32}$$

where $\text{node attention}^{(k)}$ is the node attention vector at layer k, $h_i^{(k+1)}$ is the node embedding for node i at layer $k+1$, $DP(\cdot)$ is the dropout function, $ACT(\cdot)$ is the activation function, $BN(\cdot)$ is the batch normalization function, $\omega^{(k)}$ is the trainable weight matrix for layer k, $\hat{h}_i^{(k)}$ is the normalized embedding of node i at layer k, $b^{(k)}$ is the bias term for layer k, $AGG1(\cdot)$ and $AGG_2(\cdot)$ are aggregation functions, and $\mathcal{N}(i)$ is the neighborhood of node i.

The aggregation function $AGG_1(\cdot)$ in equation (32) aggregates the transformed features of neighboring nodes with these attentions, using an element-wise product. This weighted aggregation emphasizes the contributions of neighboring nodes based on their respective attentions, thereby prioritizing nodes deemed more important. The second aggregation function, $AGG_2(\cdot)$, is the same as the $AGG(\cdot)$ in Equation (29), and it combines information from both the node itself and its neighboring effects.

For edge attention, we incorporate it through the inter-layer scheme. Specifically, we initialize a trainable $n \times 1$ edge attention vector and integrate it with the existing node representation through element-wise multiplication:

$$\hat{H}^{(k)} = \begin{cases} \left(\left(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}\right) \odot \text{edge attention}^{(k)}\right) H^{(k)}W^{(k)} & \text{if } A \text{ is symmetric} \\ \left(\left(D^{-1}\hat{A}\right) \odot \text{edge attention}^{(k)}\right) H^{(k)}W^{(k)} & \text{if } A \text{ is asymmetric} \end{cases} \tag{33}$$

where $\hat{H}^{(k)}$ is the output of the k-th layer, D is the degree matrix, $\hat{A}$ is the modified adjacency matrix, $\text{edge attention}^{(k)}$ is the edge attention vector at layer k, $H^{(k)}$ is the input to the k-th layer, $W^{(k)}$ is the

[14]The related literature includes Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin (2017), Veličković, Cucurull, Casanova, Romero, Lio, and Bengio (2017) and Thekumparampil, Wang, Oh, and Li (2018)

trainable weight matrix for the k-th layer, and $\odot$ denotes element-wise multiplication. The edge attention is updated following a standard intra-layer scheme:

$$\text{edge attention}^{(k+1)} = DP\left(ACT\left(BN\left(\text{edge attention}^{(k)}\right)\right)\right), \tag{34}$$

where $\text{edge attention}^{(k+1)}$ is the edge attention vector at layer $k+1$, and $DP(\cdot)$, $ACT(\cdot)$, and $BN(\cdot)$ are as defined previously. Under this specification in Equations (34) and (33), each node representation incorporates information from its neighboring nodes with edge attention assigned.

The integration of node and edge attention forces the neurons to put more weight on important nodes and edges to minimize the error. By outputting the weights from node and edge attention, we can open the black box of GNN to reveal which firm and which inter-firm edge is more important in the CDS prediction.

To provide a comparison, we also employ several nonlinear machine learning algorithms to predict CDS spreads, including convolutional neural network, principal component regression, partial least squares, random forest, gradient boosting regression tree and support vector regression. Consistent with GNN models, we employ the same firm-level characteristics as inputs, and the log CDS spread as the target variable. The division of the data into training, validation, and test samples is split in the same manner as in the GNN models. All hyperparameters are fine-tuned through the validation, and out-of-sample predictions are generated for the subsequent month. In this section, we discuss about the detailed algorithms of each of these machine learning models, which serve as the competing benchmarks.

D.3 CNN

We begin by illustrating the algorithm of the Convolutional Neural Network (CNN), which serves as an ideal benchmark for the Graph Neural Network (GNN). The key distinction between these two models lies in their input data: while the CNN incorporates only node characteristics, the GNN includes both node characteristics and a set of adjacency matrices. Thus, the CNN serves as a control group in our analysis, with the GNN representing a treatment group with additional edge information. In the field of machine learning, CNN is known for its powerful predictive capabilities, attributed to its complexity and nonlinearity (LeCun, Bengio, and Hinton (2015)). For instance, Gu, Kelly, and Xiu (2020) demonstrated that CNNs can predict the cross-section of stock returns more effectively than other machine learning techniques. Our CNN algorithm, designed to predict firm CDS spreads, mirrors the CNN structure in their study. Unlike the GNN architecture, the CNN architecture solely contains the updating scheme (or 'intra-layer design') and omits the inter-layer design, as there is no need to embed edge information. The intra-layer design closely resembles that of the GNN.

$$h_v^{(k+1)} = AGG\left\{ACT\left(DROPOUT\left(BN\left(\omega^{(k)} h_v^{(k)} + b^{(k)}\right)\right)\right), v = 1, 2, ..., N\right\} \tag{35}$$

The equation (35) presented in this subsection defines the update rule for the node characteristics in a Convolutional Neural Network (CNN) with one hidden layer. The node characteristics at layer $k+1$, denoted by $h_v^{(k+1)}$, are computed as a function of the characteristics of the nodes in layer k. Specifically, the

characteristics of the neighbors are aggregated using a simple linear aggregation function, and the resulting vector is passed through a sequence of modules: batch normalization $BN(\cdot)$, dropout $DROPOUT(\cdot)$, nonlinear activation function $ACT(\cdot)$, and a trainable weight matrix $\omega^{(k)}$ followed by a bias term $b^{(k)}$. The node characteristics at layer k, denoted by $h_v^{(k)}$, represent the characteristics of node v at layer k. It is not the node embedding $\hat{h}_v^{(k)}$ any more as in the GNN.

To ensure a robust comparison with the GNN, we maintain the same nonlinear activation, dropout, and batch normalization rules. The rectified linear unit (ReLU) is used as the nonlinear activation function across all nodes. Dropout is implemented to prevent overfitting, and batch normalization is employed to stabilize and expedite the training process. Therefore, the updating scheme in the CNN is identical to that in the GNN. Furthermore, we consider CNN architectures with up to four hidden layers, denoted as CNN1, CNN2, CNN3, and CNN4. The number of neurons in each layer is selected to match those in the corresponding GNN models.

Finally, the training configuration for the CNN mirrors that of the GNN. All hyperparameters, including the learning rate in SGD, are fine-tuned using the validation set, and early stopping is incorporated to prevent overfitting. In essence, all training rules for the CNN are aligned with those of the GNN, allowing for a precise comparison between the two models.

D.4 PCR and PLS

We employ Principal Components Regression (PCR) as a classic dimension reduction method in our analysis. Dimension reduction, unlike predictor selection, averages all predictors, which can help reduce noise and better isolate the signal within them. This technique is particularly beneficial when the predictors are highly correlated. PCR consists of two main steps: PCA and regression. Initially, PCA is conducted to construct a few principal components (say p components) that preserve the covariance structure of all regressors. Then these principal components are used in a standard regression model. [15] The PCR model is represented as:

$$\mathbf{CDS} = Z\beta + E \tag{36}$$

where **CDS** is the $NT \times 1$ vector of the CDS spread, Z is the $NT \times k$ matrix of stacked predictors, and E is the vector of residuals. PCR seeks the p principal components Ω so that

$$\mathbf{CDS} = (Z\Omega_p)\beta_p + \hat{E} \tag{37}$$

Equation (37) states the PCR regression after projecting predictors onto principal components.

Equation (36) states the baseline regression representation used to introduce PCR.

Each column of Ω_p, ω_j is a linear combination of the original predictors. PCR solves the following optimization function:

$$\omega_j = \arg \max_{\omega} \mathrm{Var}(Z\omega) \tag{38}$$

[15] PCA's role is to condense a large number of predictors into a few significant components before examining their relationship with the target variable.

s.t. $\omega'\omega = 1$ and $\text{Cov}(Z\omega, Z\omega_k) = 0$ fpr all $k = 1, 2, \cdots, k-1$.

Equation (38) states the variance-maximization problem that defines each PCR component.

The number of principal components is determined through the validation set. PCR seeks linear combinations of the full set of predictors that best mimic the full predictors, without directly incorporating the forecasting objective into the initial dimension reduction step. This approach sometimes leads to criticism that PCR focuses more on finding principal components than on the final prediction target.

Partial Least Squares (PLS) is another dimension reduction method that considers the correlation of initial predictors with the target during dimension reduction. For each predictor i, PLS estimates its coefficient ψ_i through a univariate OLS regression. PLS then averages all predictors into a single aggregate component, weighting them proportionally to ψ_i, thus giving more weight to stronger univariate predictors. PLS effectively averages the "partial" sensitivity of the target variable to each predictor. Multiple components can also be formed, with the target and all predictors orthogonalized with respect to previously constructed components, and the procedure repeated on the orthogonalized dataset until the desired number of PLS components is reached.

As in PCR, PLS seeks the p components Ω so that

$$\mathbf{CDS} = (Z\Omega_p)\beta_p + \hat{E} \tag{39}$$

and this number is tuned from the validation set.

Equation (39) states the analogous reduced-form regression used for PLS.

However, its optimization goal is different:

$$\omega_j = \arg \max_{\omega} \text{Cov}^2(\mathbf{CDS}, Z\omega) \tag{40}$$

s.t. $\omega'\omega = 1$ and $\text{Cov}(Z\omega, Z\omega_k) = 0$ for all $k = 1, 2, \cdots, k-1$.

Equation (40) states the covariance-maximization problem that defines each PLS component.

PLS, therefore, trades off some accuracy in principal component identification for improved prediction of the initial target.

D.5 GBRT and RF

'Ensemble' tree models are also popular in machine learning for their ability to handle nonlinearity and variable interactions. Classic tree-based models are prone to overfitting and thus require substantial regularization. Following the approach in Gu, Kelly, and Xiu (2020), we employ two regularized tree-based methods as benchmarks: Gradient Boost Regression Trees (GBRT) and Random Forest (RF).

GBRT is a method that iteratively combines predictions from multiple simple, or 'weak' trees to form a more robust, 'strong' learner. It begins by fitting a shallow tree, typically with limited depth (e.g., depth $L = 1$), which, due to its simplicity, is likely to be a weak predictor with large bias. Then a second simple tree of the same depth is used to fit the residuals from the first tree. The predictions from these two trees are combined, with the contribution from the second tree scaled down by a factor $\nu \in (0, 1)$ to mitigate overfitting. This process is repeated, with each new tree fitting the residuals from the previous ensemble,

until the ensemble comprises B trees. The final model is an additive combination of these shallow trees, with the tuning parameters (depth L, shrinkage factor ν, and number of trees B) determined during the validation phase. The optimization at each tree branch aims to minimize the L_2 loss function.

Random Forest, like GBRT, is another ensemble method that aggregates predictions from multiple trees. It is a variation of the bootstrap aggregation, or 'bagging' (see Breiman (2001)). B different bootstrap samples of the data are drawn, a separate regression tree is fitted to each sample, and their forecasts are averaged. While trees fitted to individual bootstrap samples tend to be deep and overfit, averaging across multiple predictions reduces this variability. RF introduces an additional layer of regularization by considering only a randomly selected subset of predictors for splitting at each branch, a technique akin to dropout, to further prevent overfitting. The depth L of the trees, the number of predictors considered at each split, and the number of bootstrap samples B are all key parameters, tuned through validation.

D.6 SVR

Support Vector Regression (SVR) is a machine learning method that originates from the broader concept of Support Vector Machines (SVMs). SVR is good at modeling relationships between input features and target variables. It does so by trying to optimize a hyperplane that accurately captures the underlying data patterns, while also minimizing prediction errors. SVR is grounded in a margin maximization strategy, rendering it particularly effective for both linear and non-linear regression challenges. SVR aims to identify a function $f(x)$ that predicts continuous target values from input data x. The objective is to establish a function $f(x)$ that maintains a maximal margin between the predicted values and a specified margin threshold.

Mathematically, SVR is formulated as an optimization problem. Given a training dataset $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the input feature vector of dimension d, and y_i is the corresponding target value (i.e. log CDS spread), SVR seeks to solve the following optimization problem:

$$\begin{aligned} \text{Minimize:} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) \\ \text{Subject to:} \quad & y_i - \left(w^T\phi(x_i) + b\right) \le \epsilon + \xi_i \\ & \left(w^T\phi(x_i) + b\right) - y_i \le \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0 \end{aligned} \tag{41}$$

Equation (41) states the support-vector-regression optimization problem.

Here, w is a weight vector, and b is a bias term that defines the hyperplane in the feature space. $\phi(x_i)$ represents the mapping of input data x_i into a higher-dimensional feature space. This mapping allows SVR to capture non-linear relationships between input features and target values. ϵ is a margin threshold, and ξ_i and ξ_i^* are slack variables allowing for some deviation from the margin threshold. The regularization parameter C balances maximizing the margin and minimizing the errors. [16]

The optimization problem aims to minimize the norm of the weight vector $\frac{1}{2}\|w\|^2$ while ensuring that the errors (ξ_i and ξ_i^*) are within the margin threshold ϵ. The solution yields the weight vector w and bias

[16] A smaller C encourages a wider margin but allows for more errors, while a larger C enforces stricter error penalties.

term b, and once trained, the SVR model can make predictions for new data points by evaluating $w^T \phi(x) + b$. The choice of the kernel function $\phi(x)$ plays a crucial role in SVR. We employ a polynomial kernel, with its parameters optimized through the validation set.

SVR's strength lies in its capacity to identify a margin containing most training data points, while tolerating some points to reside outside the margin within the threshold ϵ. This attribute makes SVR a robust regression method, particularly effective in scenarios involving noisy or non-linear data.

E Model Complexity and Stability

For both GNN and GNN-attention models, we use the Stochastic Gradient Descent (SGD) optimizer. To mitigate overfitting, we implement early stopping. This mechanism involves continuous monitoring of the model's performance on a separate validation dataset throughout the training process. Should there be any indication of deteriorating performance on this validation set, such as an increase in validation mean squared error, the training process is preemptively halted. This strategy is crucial for preventing the model from excessively adapting to the training data at the expense of its predictive performance on unseen data.

Additionally, we fine-tune all hyperparameters (including dropout rate, learning rate, weight decay in SGD, patience, number of hidden layers, hidden layer size of GCN and LSTM, neuron kernel initialization methods like Glorot or Kaiming, and the number of attention heads) based on the mean squared error observed on the validation set. We use simple unidirectional algorithm for LSTM and limit the number of hidden layers to either 1 or 2 for both GCN and LSTM to avoid model misspecification. The optimal configuration consists of 1 hidden layer for GCN with 12 neurons, and 2 hidden layers for LSTM with 12 and 6 neurons, respectively. For the GNN-attention model, we implement a dual-head attention mechanism and use the mean of their outputs as the final attention scores.

To calculate the number of parameters, we break down the model into its components and compute the parameters for each part. The architecture includes a GCN layer with 12 neurons, followed by a LSTM network with two layers comprising 12 and 6 neurons, respectively. The model also incorporates an attention mechanism with two attention heads. The input to the GCN layer consists of 112 variables.

First, the GCN layer applies a weight matrix to the input features, resulting in $112 \times 12 = 1,344$ parameters for the weights, plus 12 parameters for the biases, leading to a total of 1,356 parameters for the GCN layer. Next, the LSTM network requires four weight matrices for each gate (input, forget, cell, output), with dimensions based on the input size and the number of hidden units. For the first LSTM layer with 12 neurons, the number of parameters is calculated as $4 \times ((12 \times 12) + (12 \times 12) + 12) = 4 \times 300 = 1,200$ parameters. The second LSTM layer with 6 neurons requires $4 \times ((12 \times 6) + (6 \times 6) + 6) = 4 \times 114 = 456$ parameters. Finally, the attention mechanism introduces its parameters. Assuming each attention head operates on the 6-dimensional output from the last LSTM layer, and each head produces a scalar output, each attention head contributes $6 \times 1 = 6$ parameters, leading to $6 \times 2 = 12$ parameters for the two heads. Consequently, the total number of parameters in the TGNN is $1,356$ (GCN) + $1,200$ (LSTM Layer 1) + 456 (LSTM Layer 2) + 12 (Attention Heads) $= 3,024$ parameters. The total number of observation is $72 \times 678 = 48816$.

The ratio of the number of observations to the number of parameters is around 15. This ratio is generally considered a good ratio in deep learning contexts as it suggests that the model has a reasonable amount of data relative to its complexity, which can help in achieving good generalization and avoiding overfitting.

F Robustness of GNN Under Various Specifications

For robustness checks, we present out-of-sample prediction results with modified input specifications for the GNN algorithm.

In our baseline results, we construct inter-firm edge characteristics by computing idiosyncratic volatility spillover among sectors and extrapolating it to the firm level, using NAICS two-digit codes for sector classification. We now consider two alternatives: (1) Fama-French 48 sectors; and (2) Fama-French 12 sectors. The results are reported in the first two rows of Table 6.

Additionally, we explore different specifications for calculating idiosyncratic volatility. In the baseline, idiosyncratic volatility is computed using the monthly standard deviation of daily idiosyncratic returns, where idiosyncratic return is the residual after removing Fama-French 3 factors. We consider three alternatives: (1) Removing only the CAPM factor; (2) Removing PC 5 factors; and (3) Not removing any common factors. The results are reported in row 3-5 of Table 6.

Table 6. Pooled Out-of-Sample RMSE Under Various Input Specifications
This table presents the pooled out-of-sample root mean square error (RMSE) for modified input specifications for the GNN algorithm. Column 1 shows RMSE for the entire sample from February 2005 to December 2020. Columns 2 and 3 report RMSE for investment-grade (BBB and above) and high-yield (BB and below) firms, respectively. Columns 4 and 5 display RMSE for small (below median market capitalization) and large (at or above median market capitalization) firms, respectively.

	All	Investment Grade	High Yield	Small	Big
Fama-French 48 sectors	0.850	0.775	0.977	0.822	0.816
Fama-French 12 sectors	0.820	0.726	0.974	0.800	0.770
CAPM factor	0.802	0.709	0.877	0.878	0.865
PC 5 factors	0.701	0.704	0.697	0.629	0.637
no factor	0.902	0.714	1.079	0.888	0.733

Overall, these robustness checks confirm the reliability and effectiveness of the edge characteristics design in our GNN algorithms, demonstrating their robustness across various specifications.