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

Jonathan Brogaard† Belinda (Chen) Chen ‡

Abstract

We develop a structural framework of credit counterparty risk in which contagion arises from an inter-firm production network. Empirically, we propose a network-based machine learning approach that directly incorporates network topology to predict credit spreads. We find that incorporating network edge features induces an average credit-spread change of approximately 21.8% and yields an incremental R^2 of 0.56 in explaining credit spreads. Our results show that network-based counterparty risk is strongly priced and plays a first-order role in shaping credit spreads, particularly during periods when production networks experience severe disruption, reorganization, or rewiring. Network effects are especially important for firms operating in industries that occupy intermediate positions within supply chains, rely heavily on distribution and logistics, and face low substitutability of inputs.

Keywords: Credit Risk, Counterparty 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), Dana Kiku, 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 intensities constitutes a central topic in the study of credit risk. A large body of literature documents that default clustering arises for multiple reasons. First, firms are exposed to common shocks or correlated risk factors (e.g., Duffie and Singleton (1999), Collin-Dufresne, Goldstein, and Martin (2001), and Duffie and Gârleanu (2001)), so that default probabilities are jointly driven by aggregate macroeconomic conditions. Second, firms may experience firm-specific (idiosyncratic) shocks that propagate to other firms due to economic linkages, giving rise to contagion through firm networks.

Empirical studies such as Das, Duffie, Kapadia, and Saita (2007) and Jorion and Zhang (2009) provide strong evidence that idiosyncratic risk contagion—often referred to as counterparty risk—can explain excess default clustering beyond what is attributable to common risk factors alone. In addition, a growing theoretical literature (e.g., Jarrow and Yu (2001), Jorion and Zhang (2007), and Azizpour, Giesecke, and Schwenkler (2018)) has developed structural frameworks that incorporate counterparty risk into models of default intensities and credit spreads. However, due to substantial analytical and estimation complexity, most existing models are tractable only under highly simplified settings, such as two-firm economies or assumptions that rule out cascading or looping effects—that is, situations in which a firm-specific shock propagates through multiple firms and potentially feeds back into the originating firm before spreading further through the network.

In this paper, we study a general credit risk framework with the following features: (i) n firms are economically linked through a network structure; (ii) cascade effects are permitted; (iii) contagion is asymmetric across firms; and (iv) ordinary idiosyncratic shocks—rather than only extreme firm-specific tail events—can propagate through the network. Our analysis focuses on the implications of the inter-firm network structure for the cross-sectional behavior of credit spreads.

Unlike much of the existing literature in which economic linkages are left unspecified, we explicitly model inter-firm connections as production-based input–output relationships. This discipline allows us to provide a granular network-based interpretation of counterparty risk, clarifying how both the global structure of the network and a firm's position within it shape the propagation of idiosyncratic default risk. In practice, firms are connected through multiple channels, including production networks, cross-holdings of assets, debt and liability exposures, and other contractual relationships, all of which may facilitate idiosyncratic risk spillovers. We focus on the production network as the primary channel because there is substantial

empirical evidence that real default contagion occurs along supplier–customer chains (e.g., Jacobson and Von Schedvin (2015)). While we do not rule out alternative transmission mechanisms, our framework emphasizes the production network to deliver a clear and economically grounded mechanism. Once this channel is well understood, the analysis can be readily extended to incorporate other forms of inter-firm linkages. Finally, our framework does not rely on specific time-series assumptions about default intensities.[1] Instead, our primary contribution lies in characterizing how the cross-sectional $n \times n$ network structure governs the propagation of idiosyncratic default risk.

Theoretically, we develop a structural framework in which each firm's default intensity depends on a macroeconomic variable M, firm-specific characteristics X_i, and counterparty risk components a_{ij}, where a_{ij} captures idiosyncratic default-risk spillovers from firm j to firm i through the production input-output network. We provide an economic interpretation of a_{ij} by building on the key insights of Herskovic (2018) and Chen (2023). Specifically, both studies develop a production-based general equilibrium model with n representative firms that links real production to firms' investment decisions. In this framework, each firm produces a single good that is used as an intermediate input by other firms, and firms use sales revenue—that is, cash flows—to issue equity held by households. Firms are subject to i.i.d. idiosyncratic productivity shocks, which affect their own final output and subsequently spill over to downstream firms by affecting input supplies, cash flows, and stock returns.[2]

Motivated by this result, we define an $n \times n$ adjacency matrix $A \equiv [a_{ij}]$ to characterize the structure of *idiosyncratic* risk spillovers arising from the production network. As a model implication, each element a_{ij} is an endogenous function of the underlying production network weights. A larger value of a_{ij} indicates that firm j exerts a stronger idiosyncratic risk propagation effect on firm i. This effect reflects the overall importance of firm j to firm i through three channels: (i) the direct importance of firm j's output as an input supplier to firm i; (ii) the indirect importance of firm j's output through other firms in the production network that subsequently affect firm i; and (iii) the importance of firm j's output for aggregate household consumption, which in turn feeds back to firm i.

When a_{ij} is large, firm j constitutes an economically important supplier to firm i, capturing both direct and indirect input linkages. A negative idiosyncratic shock to firm j reduces its output and disrupts firm i's

[1] Das et al. (2007) adopt a similar modeling strategy.

[2] While both Herskovic (2018) and Chen (2023) do not distinguish between positive and negative idiosyncratic productivity shocks, we focus on the propagation of *negative idiosyncratic shocks* and interpret these spillovers specifically in terms of default risk.

effective input supply. This disruption lowers firm i's expected cash flows and equity value, while increasing the volatility of cash flows and, consequently, equity return volatility.

We link this channel to firm i's default intensity by adopting a distance-to-default (DD) framework following Bharath and Shumway (2008). In this framework, the original Merton DD model is simplified in a computationally tractable manner, allowing all components to be approximated using observable market data. Specifically, a firm's distance to default depends on the deviation of the market value of equity from the face value of debt, scaled by the volatility of firm asset value (the combined value of debt and equity), where asset volatility can be approximated as a linear function of equity volatility.

Under the assumption of a constant interest rate and fixed debt face value and maturity, a decline in firm i's equity value and an increase in firm i's equity volatility both reduce its distance to default. A lower distance to default, in turn, implies a higher default probability and default intensity. Consequently, a larger a_{ij} increases firm i's default intensity through the cash-flow news channel operating via the input-output linkage.

After expressing firm i's default intensity as a structural function that depends positively on the network components a_{ij}, we derive the credit default swap (CDS) spread for firm i under the risk-neutral measure. The CDS spread is determined by equating the expected present value of premium payments to the expected present value of default loss payments. The analytical solution implies that firm i's CDS spread depends on a *nonlinear* function of macroeconomic conditions, firm-specific characteristics, and production-network spillover terms a_{ij} for all j, with the partial derivative $\frac{\partial \text{CDS}_i}{\partial a_{ij}} > 0$. This result implies that credit spreads are best explained by incorporating the production-network structure $\{a_{ij}\}$ in a nonlinear manner.

We next provide empirical evidence to support the structural framework. There are two natural approaches, both of which face significant obstacles. First, directly estimating the fully specified structural model is computationally infeasible in large production networks. Existing structural counterparty-risk models, such as Jarrow and Yu (2001), are tractable only under highly simplified assumptions—for example, two-firm economies without cascading network effects. Second, directly applying statistical prediction methods, including standard machine-learning algorithms, faces a fundamental limitation relative to our structural framework: these approaches typically rely on firm-level features as inputs, but cannot directly incorporate the full network topology—represented by an $n \times n$ adjacency matrix—as a predictive object.

To address these challenges, we adopt a recently developed machine-learning methodology: graph neural networks (GNNs). GNNs are explicitly designed to capture global graph topology and to model data

with complex cross-sectional dependence, making them a natural empirical counterpart to our production-network-based structural framework.

Unlike much of the machine-learning literature, which emphasizes extensive model tuning and increasingly sophisticated architectures to maximize pooled out-of-sample predictive performance, our objective is different. We deliberately employ a parsimonious GNN architecture that is sufficient to deliver clear financial interpretation of network effects. In doing so, we sacrifice architectural complexity in favor of economic transparency. Our empirical analysis pursues two primary objectives: (i) to assess the incremental predictive power of production-network information for credit spreads relative to non-network benchmarks, and (ii) to characterize which firms' credit spreads are most exposed to counterparty risk through the network and the economic states in which such network risk is most strongly priced.

In our GNN framework, each firm is represented as a node, and each firm-to-firm idiosyncratic risk spillover is represented as a weighted, directed edge. The model incorporates both node-level features—such as firm-specific characteristics and macroeconomic variables—and edge-level features, which capture the magnitude of production-network spillovers a_{ij}. These spillover measures are estimated using input–output data and stock return data following Diebold and Yılmaz (2014) and Chen (2023). The GNN aggregates information from neighboring nodes and edges to construct firm-level latent representations, which are then used to predict CDS spreads.

We evaluate the empirical results in light of the two objectives above by conducting the following analyses. First, we compare the out-of-sample performance of the GNN model with a set of nonlinear benchmark models that rely on the same node-level features but exclude network information. Among these benchmarks, the convolutional neural network (CNN) provides a natural point of comparison. We design the CNN architecture under strict assumptions so that it corresponds to a GNN in which all network edges are set to zero.

Under identical training and validation protocols, we find that the GNN substantially reduces out-of-sample prediction error, with the RMSE declining to 0.89, compared with 1.34 for the CNN benchmark. Economically, we find that incorporating network edge features induces an average spread change of approximately 21.8% in the full sample and yields an incremental R^2 of 0.56 in explaining credit spreads. This result corroborates the central role of network effects in the structural model.

The improvement in predictive accuracy is present for both investment-grade and high-yield firms, with a more pronounced effect for investment-grade firms. Intuitively, investment-grade firms are typically

large, mature, and highly interconnected through multiple input–output relationships, which places them in more central positions within production networks. As a result, incorporating network characteristics is particularly informative for predicting the CDS spreads of investment-grade firms, whose credit risk is more exposed to contagion through complex production linkages.

Second, we assess the time-varying importance of network structure by examining out-of-sample prediction performance at a monthly frequency. The resulting time-series evidence reveals that GNN models consistently outperform all competing approaches throughout the sample. In several periods, the RMSE achieved by GNNs is as low as one quarter of that produced by alternative algorithms, particularly relative to dimension-reduction methods such as PCR and PLS. This persistent performance gap indicates that pairwise inter-firm network information captures a substantial component of credit risk that is not explained by firm-level characteristics alone.

Moreover, we find there are several episodes in which network structure becomes especially important. The first occurs during the 2008 financial crisis. During severe economic downturns, production networks are disrupted as firms face liquidity constraints and operational distress. Under such conditions, inter-firm contagion and counterparty risk become first-order determinants of credit spreads, which GNNs are well equipped to capture.

A second episode occurs in late 2009 and early 2010, when equity prices rebounded sharply and firms began repairing previously disrupted production linkages. This transition period involved substantial uncertainty due to portfolio reallocation and network reconfiguration. Models that ignore network structure experience sharp increases in RMSE, whereas GNNs maintain strong predictive performance by accounting for evolving inter-firm dependencies.

A third episode occurs around 2018, coinciding with the escalation of trade tensions and tariff policies. Tariffs can be viewed as disruptions to trade linkages that increase fragility in global value chains. During this period, CDS spreads reflect firms' exposure to affected trade partners, again amplifying the role of network structure.

Finally, following the onset of the COVID-19 pandemic in 2020, widespread production shutdowns—such as factory closures in the semiconductor sector—generated severe upstream supply disruptions. Downstream firms reliant on these inputs experienced heightened fragility, leading to pronounced network-driven credit risk. GNNs capture these cascading effects, while non-network models fail to do so.

The time-series evidence suggests that the full inter-firm network captures an important component of

systematic risk that is priced in CDS spreads. These findings echo the production network literature (e.g., Acemoglu, Carvalho, Ozdaglar, and Tahbaz-Salehi, 2012), which emphasizes that network structure provides a microfoundation for systematic risk.

In addition, we examine the network-attributable credit spread change across sectors to identify which industries' credit spreads are more susceptible to supply-chain disruptions.

We find that network features tend to matter more for firms that (i) occupy intermediate positions in the production chain and are exposed to both upstream and downstream risks, (ii) rely heavily on shipping, logistics, and distribution across the supply chain, or (iii) depend on specialized inputs that are difficult to substitute. For such firms, disruptions or shocks propagating through supplier–customer relationships can lead to substantial changes in expected cash flows and, consequently, credit spreads. Industries exhibiting these characteristics include electrical equipment, shipbuilding, railroad equipment, and defense. For example, we find that network features induce especially large spread changes for the defense sector during the period from 2014 to 2018, which coincides with discrete geopolitical shocks such as the Russia–Ukraine conflict and conflicts involving ISIS. During such periods, defense demand becomes lumpy and state-dependent, and contracts may be renegotiated, accelerated, delayed, or repriced.

By contrast, industries whose credit risk is primarily driven by firm-specific fundamentals or final demand exhibit smaller network-attributable spread changes. For instance, network edge features induce relatively modest shifts in the credit spreads of natural resource extraction industries (mining and coal), which are heavily influenced by commodity prices and balance-sheet leverage. Similarly, the entertainment and textile sectors rely more heavily on idiosyncratic fundamentals such as intellectual property, brand value, and consumer sentiment, and often operate with more diversified supplier and customer bases. As a result, changes in supply-chain networks play a more limited role in shaping their credit spreads.

Related Literature

This paper contributes to the literature on credit risk. A large body of work proposes structural models to study default clustering, credit risk spillovers, and interconnectedness in credit markets. Default clustering is commonly attributed to two broad mechanisms. First, firms may be exposed to common risk factors, which are typically modeled through state variables or aggregate macroeconomic shocks (e.g., Collin-Dufresne, Goldstein, and Martin (2001), Duffie and Gârleanu (2001)). Second, default clustering may arise from firm-specific shocks—namely, idiosyncratic risks—that propagate through economic linkages. Related studies

include Jarrow and Yu (2001), Das et al. (2007), Jorion and Zhang (2007), Jorion and Zhang (2009), and Azizpour, Giesecke, and Schwenkler (2018). Our paper primarily contributes to the second strand of the literature by studying idiosyncratic default-risk contagion through inter-firm networks. To our knowledge, this is the first paper to provide a comprehensive framework for understanding how the entire network structure shapes credit risk. While most existing structural models focus on highly simplified settings—such as two-firm economies, the absence of cascading effects, and contagion driven only by extreme firm-specific tail events—and typically do not explicitly specify the underlying economic linkages, we develop a more general credit risk structural model with the following features: (i) n firms are economically linked through a network structure that is explicitly identified as a production-based input–output network; (ii) cascade effects are permitted; (iii) contagion is asymmetric across firms; and (iv) ordinary idiosyncratic shocks, rather than only tail events, can propagate through the network. We focus on examining how inter-firm network structure and firms' positions within the network shape the cross-sectional behavior of credit spreads.[3] While some CDS-focused studies (e.g., Getmansky, Girardi, and Lewis (2016)) emphasize counterparty risk arising from common protection sellers such as dealers, our analysis abstracts from dealer cores and OTC market structure. Instead, we focus on inter-firm economic connections operating through production networks.

Beyond the counterparty-risk literature, there is a broad body of research on modeling and measuring credit and default risk, including 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). We contribute to this literature by explicitly incorporating counterparty risk through inter-firm networks into a structural credit risk framework.[4] Some papers (e.g., Giesecke, Longstaff, Schaefer, and Strebulaev (2011)) emphasize the presence of a non-default-related credit risk premium. From a structural perspective, this premium compensates investors for exposure to adverse states that do not necessarily involve realized default. Such premia are particularly important during recessions, when bond prices decline sharply, and are often modeled through time-varying volatility, differences between physical and risk-neutral default probabilities, or rare-disaster risk. In contrast, our paper focuses on a parsimonious framework that does not

[3]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).

[4]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).

rely on specific time-series assumptions about default intensities—an approach also adopted by Das et al. (2007). Instead, we emphasize how the cross-sectional $n \times n$ network structure governs the propagation of default risk.

Our work also contributes to the growing literature on inter-firm networks, particularly production-based networks. A large body of research develops production-based multisector models to study the implications of network structure for aggregate productivity, aggregate volatility, and macroeconomic tail risk (e.g., Carvalho (2008), Acemoglu et al. (2012), Carvalho and Gabaix (2013), Herskovic (2018), and Chen (2023)). We adapt this production-based input–output framework to provide an economically grounded explanation for the existence of counterparty risk and to offer a comprehensive analysis of the role of production networks in shaping credit spreads.[5]

Finally, this paper contributes to the growing literature on machine learning in finance. A number of studies, including Kelly, Pruitt, and Su (2019), Gu, Kelly, and Xiu (2020), and Gu, Kelly, and Xiu (2021), apply machine-learning techniques to asset pricing and return prediction, demonstrating their effectiveness in handling high-dimensional data. Our paper introduces graph neural networks (GNNs) as a novel empirical tool for integrating granular and global network-topology information to explain credit spreads.[6]

The remainder of the paper is organized as follows. Section 2 outlines a conceptual framework for CDS pricing. Section 3 presents the empirical evidence. Section 4 concludes.

2 Structural Framework

2.1 Production-Network Exposure

We consider a production-based multi-firm network model developed in Herskovic (2018) and Chen (2023), which links real production to firms' investment and asset prices. While these frameworks abstract from leverage and default, we extend the setting by allowing firms to issue defaultable debt in the form of a zero-coupon bond.

We study a production economy with n firms indexed by $i = 1, \ldots, n$. Each firm purchases other firms'

[5] 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).

[6] 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).

goods as intermediate inputs in order to produce its own output. A firm's output can either be used as an intermediate input by other firms or be directly consumed by the representative household. Each firm's production technology is subject to negative idiosyncratic productivity shocks.

Firms use the cash flows generated from selling their real output to issue equity, which is held by the household. As a result, idiosyncratic productivity shocks propagate through the supply chain and affect firms' cash flows, dividends, and stock returns. In addition, each firm i issues a single zero-coupon bond with face value F_i and maturity T. The risk-free interest rate r is constant, and the face value F_i is fixed and does not vary with economic conditions.

As shown in Chen (2023), the production network generates rich cross-firm dependencies not only in cash flows, dividends, stock returns, and return volatilities, but also in idiosyncratic return volatilities—that is, the volatility of residual returns after removing common components. Importantly, this idiosyncratic volatility spillover structure provides a natural measure of pairwise systemic risk within the production network. In this paper, we focus on this pairwise systemic risk measure and explore its implications for credit spreads. Specifically, we define an $n \times n$ adjacency matrix,

$$A \equiv [a_{ij}]_{i,j=1}^{n}, \tag{1}$$

which characterizes pairwise idiosyncratic risk spillovers across firms as summarized by equation (1). We take this structural result as the starting point of our analysis and leave the detailed derivation of the full production equilibrium to the Internet Appendix.

As a model implication, each element a_{ij} in equation (1) is an endogenous function of the underlying production network weights. A larger value of a_{ij} indicates that firm j exerts a stronger idiosyncratic risk propagation effect on firm i. This effect reflects the overall importance of firm j to firm i through three channels: (i) the direct importance of firm j's output as an input supplier to firm i; (ii) the indirect importance of firm j's output through other firms in the production network that subsequently affect firm i; and (iii) the importance of firm j's output for aggregate household consumption, which in turn feeds back to firm i.

Let $u_{j,t} \geq 0$ denote a negative idiosyncratic productivity shock to firm j at time t. Firm i's operating cash-flow innovation is given by

$$\Delta\pi_{i,t} = \beta_M' M_t + \beta_X' X_{i,t} - \sum_{j=1}^{n} a_{ij} u_{j,t}, \tag{2}$$

as shown in equation (2), where M_t is a vector of aggregate macroeconomic conditions and $X_{i,t}$ is a vector of firm-specific characteristics.

Negative supplier shocks reduce firm i's expected cash flows in proportion to its downstream exposure a_{ij} in equation (2). In addition, supplier disruptions increase uncertainty in downstream production. We therefore allow cash-flow volatility to depend on production-network exposure according to

$$\mathrm{Var}_t(\Delta\pi_{i,t}) \equiv \sigma^2_{\pi,i,t} = \bar{\sigma}^2_{\pi,i} + \sum_{j=1}^{n} a^2_{ij}\sigma^2_{u,j,t}, \tag{3}$$

as specified in equation (3), where $\bar{\sigma}^2_{\pi,i}$ is a firm-specific baseline variance and $\sigma^2_{u,j,t}$ denotes the conditional variance of supplier j's downside shock.

2.2 Capital Structure and Distance to Default

Let $V_{i,t}$ denote the market value of firm i's assets at time t, defined as the present value of future operating cash flows. Equity holders have limited liability, so the market value of equity satisfies

$$E_{i,t} = \max\left(V_{i,t} - F_i e^{-r(T-t)},\, 0\right), \tag{4}$$

as given in equation (4).

Following the distance-to-default framework of Bharath and Shumway (2008), distance to default for firm i at time t is defined as

$$DD_{i,t} = \frac{\ln\left(\frac{V_{i,t}}{F_i}\right) + \left(r - \frac{1}{2}\sigma^2_{V,i,t}\right)(T-t)}{\sigma_{V,i,t}\sqrt{T-t}}, \tag{5}$$

as shown in equation (5), where $\sigma_{V,i,t}$ denotes the volatility of firm asset value.

Consistent with Bharath and Shumway (2008), asset value and asset volatility are approximated using observable equity market data according to

$$V_{i,t} \approx E_{i,t} + F_i, \tag{6}$$

$$\sigma_{V,i,t} \approx \frac{E_{i,t}}{V_{i,t}}\sigma_{E,i,t}, \tag{7}$$

as specified in equations (6) and (7), where $\sigma_{E,i,t}$ denotes equity return volatility.

We allow equity value and equity volatility to depend on production-network exposure as follows:

$$\ln E_{i,t} = \alpha_i + \alpha_M' M_t + \alpha_X' X_{i,t} - \sum_{j=1}^{n} a_{ij}\theta_{j,t} + \eta_{i,t}, \tag{8}$$

$$\sigma_{E,i,t} = \bar{\sigma}_{E,i} + \sum_{j=1}^{n} a_{ij}\phi_{j,t}, \tag{9}$$

as described in equations (8) and (9), where $\theta_{j,t} \geq 0$ captures supplier-specific downside cash-flow news and $\phi_{j,t} \geq 0$ captures supplier-specific downside uncertainty.

2.3 Default Probability and Credit Spreads

To price credit default swaps, we approximate the risk-neutral default probability over the CDS horizon $[t, T]$ as

$$\mathbb{P}^{\mathbb{Q}}_{i,t}(\text{default by } T) \approx \Phi\big(-DD_{i,t}\big)\,, \tag{10}$$

as given in equation (10), abstracting from time variation in risk premia.

Assuming fractional recovery of par $R \in (0, 1)$ and a zero-coupon approximation for the CDS contract, the par CDS spread $s_{i,t}$ satisfies

$$s_{i,t} = (1 - R)\frac{\mathbb{P}^{\mathbb{Q}}_{i,t}(\text{default by } T)}{T - t}, \tag{11}$$

as defined in equation (11).

Substituting equation (10) into equation (11) yields the simplified CDS spread,

$$s_{i,t} = \frac{1 - R}{T - t}\,\Phi\big(-DD_{i,t}\big)\,, \tag{12}$$

as shown in equation (12).

Equations (1)–(12) jointly imply that firm i's CDS spread is a nonlinear function of macroeconomic conditions, firm-specific characteristics, and production-network exposure $\{a_{ij}\}_{j=1}^{n}$. In particular, greater downstream exposure to negative supplier shocks lowers distance to default and increases credit spreads.

3 Empirical Evidence

We provide empirical support for our structural framework, which emphasizes the importance of the entire network topology in explaining credit spreads. To this end, we adopt a recently developed machine-learning methodology—graph neural networks (GNNs)—to conduct CDS spread prediction exercises. GNNs are specifically designed to model data with complex cross-sectional dependence and to incorporate rich network topology features as inputs, making them a natural empirical counterpart to our structural framework.

The basic intuition behind the GNN architecture is to represent the inter-firm network explicitly. Each firm is modeled as a node, while each firm-to-firm idiosyncratic risk spillover measure is represented as a weighted and directed edge. The model takes both node-level features—such as firm-specific characteristics and macroeconomic variables—and edge-level features, namely the idiosyncratic risk spillover measures a_{ij}, as inputs. Through a sequence of message-passing and aggregation steps, the GNN embeds information from the network structure and edge weights into latent node-level representations, which are then used to predict CDS spreads in a standard neural network framework.

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 edge-level features, and finally present the CDS spread prediction results.

3.1 Architecture of the Empirical Algorithm

In much of the machine-learning literature, researchers emphasize 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 consists of two key components: an inter-layer message-passing scheme and an intra-layer updating scheme, with the latter corresponding to the standard updating mechanism 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

This scheme determines how initial node-level and edge-level features are transformed into latent node representations. Specifically, each normalized embedding $\hat{h}_i^{(k)}$ at layer k incorporates information from node i and its neighbors according to

$$\hat{h}_i^{(k)} = h_i^{(k)} \bigoplus_{j \in \mathcal{N}(i)} (a_{ij}), \tag{13}$$

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} represents 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). Under this framework, network information is incorporated as

$$\hat{H}^{(k)} = D^{-1}\hat{A}H^{(k)}W^{(k)}, \quad \text{if } A \text{ is asymmetric}, \tag{14}$$

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.

The intuition is as follows. Each row of the adjacency matrix $A = [a_{ij}]$ reflects the extent to which firm i receives idiosyncratic risk spillovers from other firms. By multiplying this row by neighboring firms' features and aggregating the results, the model assigns greater weight to more economically important suppliers. After aggregating weighted supplier 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. Thus, the contribution of neighboring information depends on the economic importance of the supplier to firm i.

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. This choice is beneficial 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.[7] 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.[8]

We construct a monthly panel by retaining the most recent CDS spread observed in each month. The final dataset contains 678 firms and 130,176 firm-month observations over the sample period.

TABLE 1 AND FIGURE 3 ABOUT HERE

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

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

[8] 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.

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.[9] 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.[10]

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).[11] 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 $[-1, 1]$.

Edge Characteristics. To construct edge-level features, we estimate pairwise idiosyncratic volatility spillovers using CRSP daily stock returns, consistent with the structural framework. This procedure follows Chen (2023), which generalizes the systemic risk measures developed in Diebold and Yılmaz (2014). We assume that firms within the same industry produce similar products and are subject to similar productivity-specific shocks. Accordingly, we estimate risk spillovers at the sector level and impose them on firms based on industry classification.

[9] 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.

[10] 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.

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

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.[12] 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.[13] This procedure yields a 48×48 matrix of spillover intensities, where each element a_{ij} measures the contribution of shocks originating in sector j to the idiosyncratic volatility of sector i. These measures constitute the empirical counterpart of the structural spillover weights a_{ij} in the model.

We collect the estimated a_{ij} from each rolling window to form a time-varying adjacency matrix A_t updated at a monthly frequency.[14]

Extrapolation to the Firm Level. After obtaining sector-level spillover measures, we extrapolate them to the firm level by imposing sectoral risk 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 intensity from any firm in sector 2 to any firm in sector 1.

By imposing sector-level adjacency matrices at the firm level, we map inter-sectoral dependencies into inter-firm dependencies. This approach implicitly 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 our production-network framework, in which firms within a sector face similar productivity shocks. For clarity, we abstract from additional firm-specific shock channels.

Finally, one may ask why we do not construct idiosyncratic volatility spillovers directly from CDS data.

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

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

[14] 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.

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. Accordingly, 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.

TABLE 2 AND FIGURE 4 ABOUT HERE

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:

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

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:

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

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 inter-firm 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 inter-firm 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).

These patterns are economically intuitive. Network features tend to matter more for firms that (i) occupy intermediate positions in the production chain and are exposed to both upstream and downstream risks, (ii) rely heavily on shipping, logistics, and distribution across the supply chain, or (iii) depend on specialized inputs that are difficult to substitute. For such firms, disruptions or shocks propagating through supplier–customer relationships can lead to substantial changes in expected cash flows and, consequently, credit spreads.

The industries highlighted in red—notably electrical equipment, shipbuilding, railroad equipment, and defense—are precisely those that lie at critical intermediate nodes of the production and trade network. Their

revenues depend on continuous flows from multiple counterparties, making them particularly sensitive to network-level disruptions. Defense-related firms, in particular, rely on highly specialized and sophisticated components sourced through rigid supply chains and often operate under long-term contracts. This network rigidity, characterized by low input substitutability, amplifies the importance of network edge information in determining the credit health of the defense sector. Consistent with this interpretation, network features induce especially large spread changes for defense sector during the period from 2014 to 2018, which coincides with discrete geopolitical shocks such as the Russia–Ukraine conflict and conflicts involving ISIS. During such periods, defense demand becomes lumpy and state-dependent, and contracts may be renegotiated, accelerated, delayed, or repriced.

By contrast, the industries highlighted in blue are those whose credit risk is primarily driven by firm-specific fundamentals or final demand. For example, firms in natural resource extraction (mining and coal) are heavily influenced by commodity prices and balance-sheet leverage, which dominate network considerations. Similarly, entertainment and textile firms rely more heavily on idiosyncratic fundamentals such as intellectual property, brand value, and consumer sentiment, and often operate with more diversified supplier and customer bases. As a result, network edge features induce relatively smaller shifts in their credit spreads.

4 Conclusion

This paper develops a structural framework for credit counterparty risk in which idiosyncratic shocks propagate through an inter-firm production network. By explicitly modeling firms' economic linkages as input–output relationships, the framework clarifies how both a firm's position within the network and the global network topology shape the transmission of idiosyncratic default risk and the cross-sectional behavior of credit spreads. The model allows for asymmetric contagion, cascading effects, and ordinary firm-specific shocks, and links production-network spillovers to default intensities and CDS spreads through a distance-to-default channel.

To provide empirical support for the structural mechanism, we propose a parsimonious network-based empirical approach that directly incorporates the full inter-firm network topology. Using graph neural networks (GNNs), we embed both firm-level characteristics and pairwise idiosyncratic risk spillovers into a unified prediction framework. This approach overcomes a key limitation of standard empirical and machine-

learning methods, which typically rely on firm-level features and cannot incorporate the entire $n \times n$ network structure as a predictive object.

Empirically, we find that models incorporating production-network information substantially outperform 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 transitions, and supply-chain disruptions. These findings indicate that inter-firm network structure captures an important component of credit risk that is not explained by firm fundamentals or common shocks alone, and that this network-based risk is state-dependent and priced in CDS markets.

Overall, our results provide direct empirical support for a production-network-based view of counterparty risk. They suggest that credit spreads reflect not only firms' standalone default risk, but also their exposure to idiosyncratic shocks originating elsewhere in the network. More broadly, the paper demonstrates how combining structural economic modeling with modern network-based machine-learning tools can yield new insights into credit risk, contagion, and systemic risk. An important direction for future research is to extend the framework to incorporate other forms of inter-firm linkages—such as financial exposures or dealer networks—and to study the dynamic evolution of network risk over time.

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).

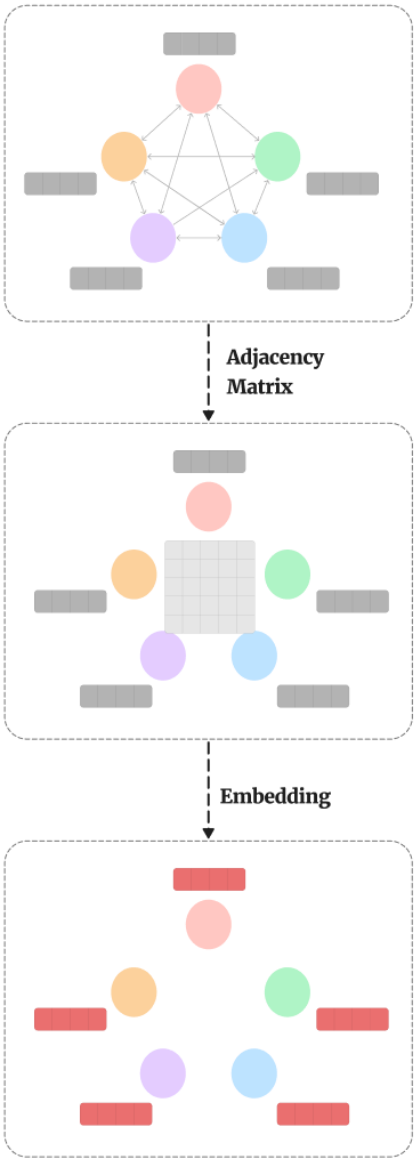
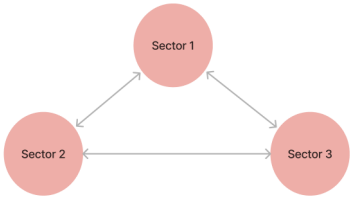


Figure 2. Extrapolation of Sector-Level Network Effects to Firm Level

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.

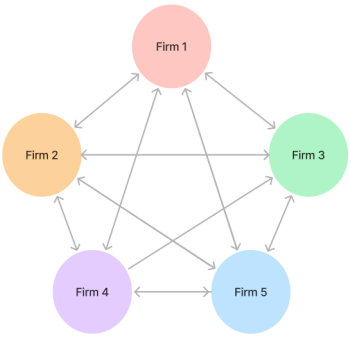
Panel A: Example for risk spillover among 3 sectors



	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.

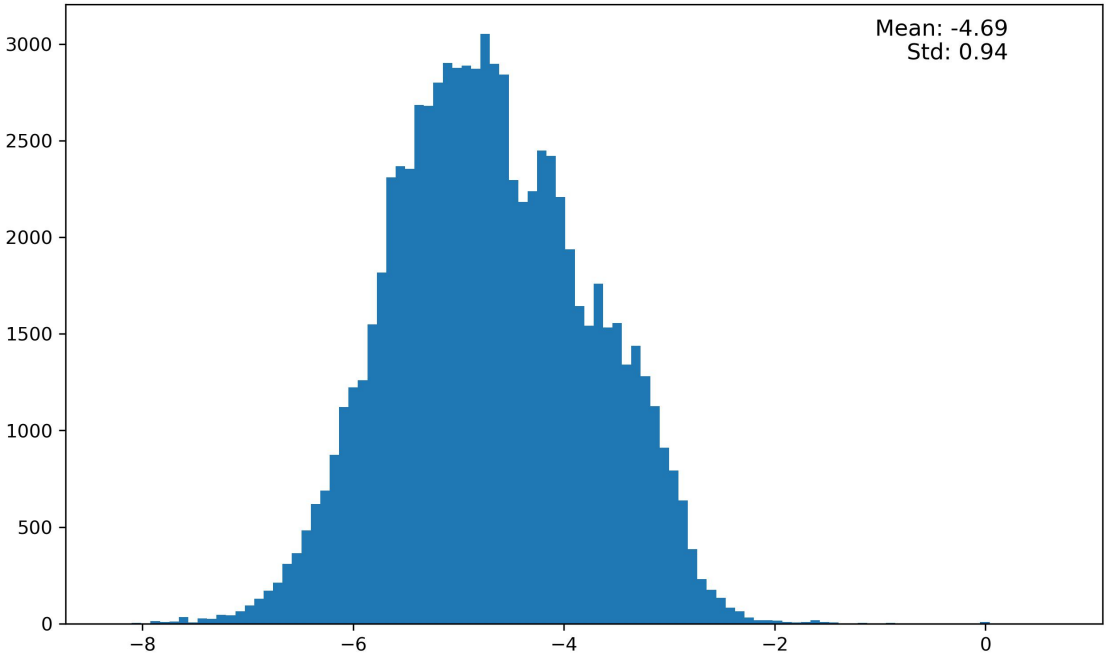
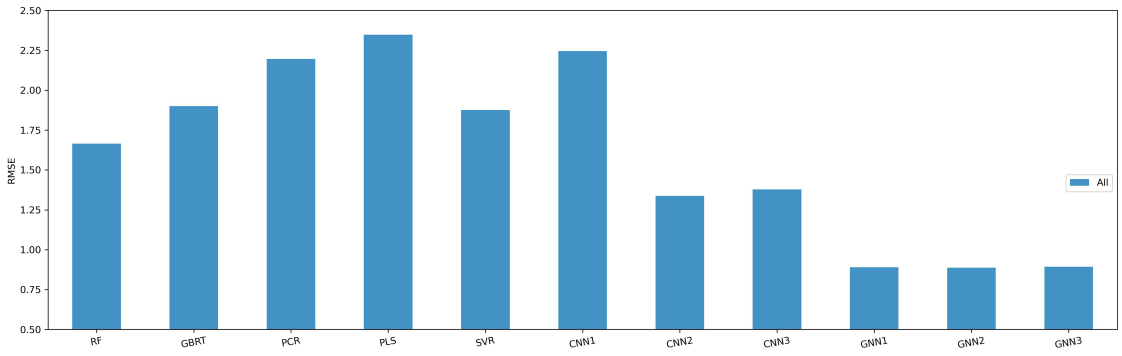
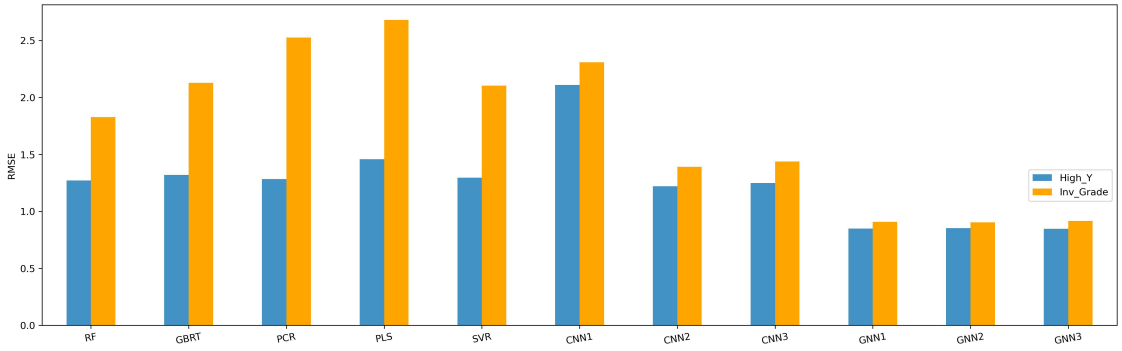


Figure 4. Out-of-Sample Prediction Error of Machine Learning Algorithms

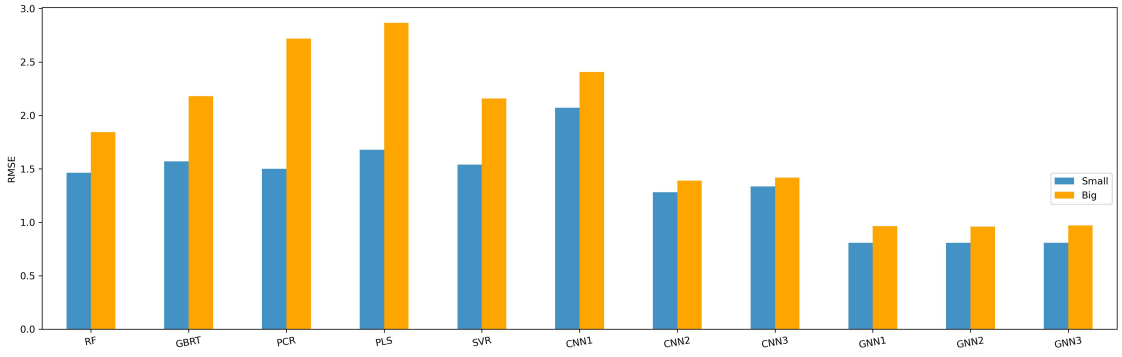
This figure presents the out-of-sample prediction error of various machine learning algorithms using pooled Root Mean Square Error (RMSE). Panel A displays the overall RMSE across all out-of-sample periods. Panel B presents the overall RMSE by firm credit rating, with investment grade defined as BBB 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.

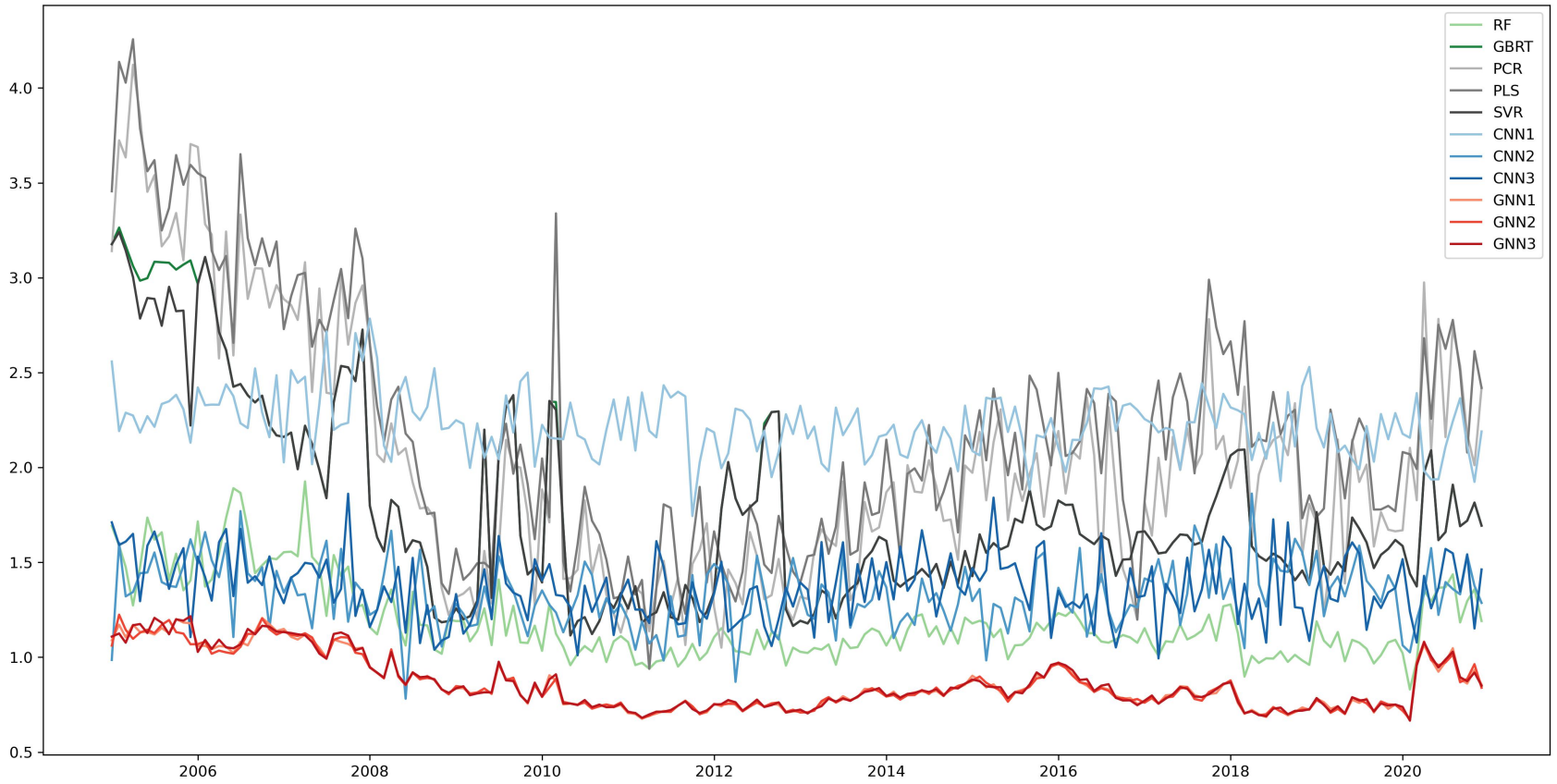
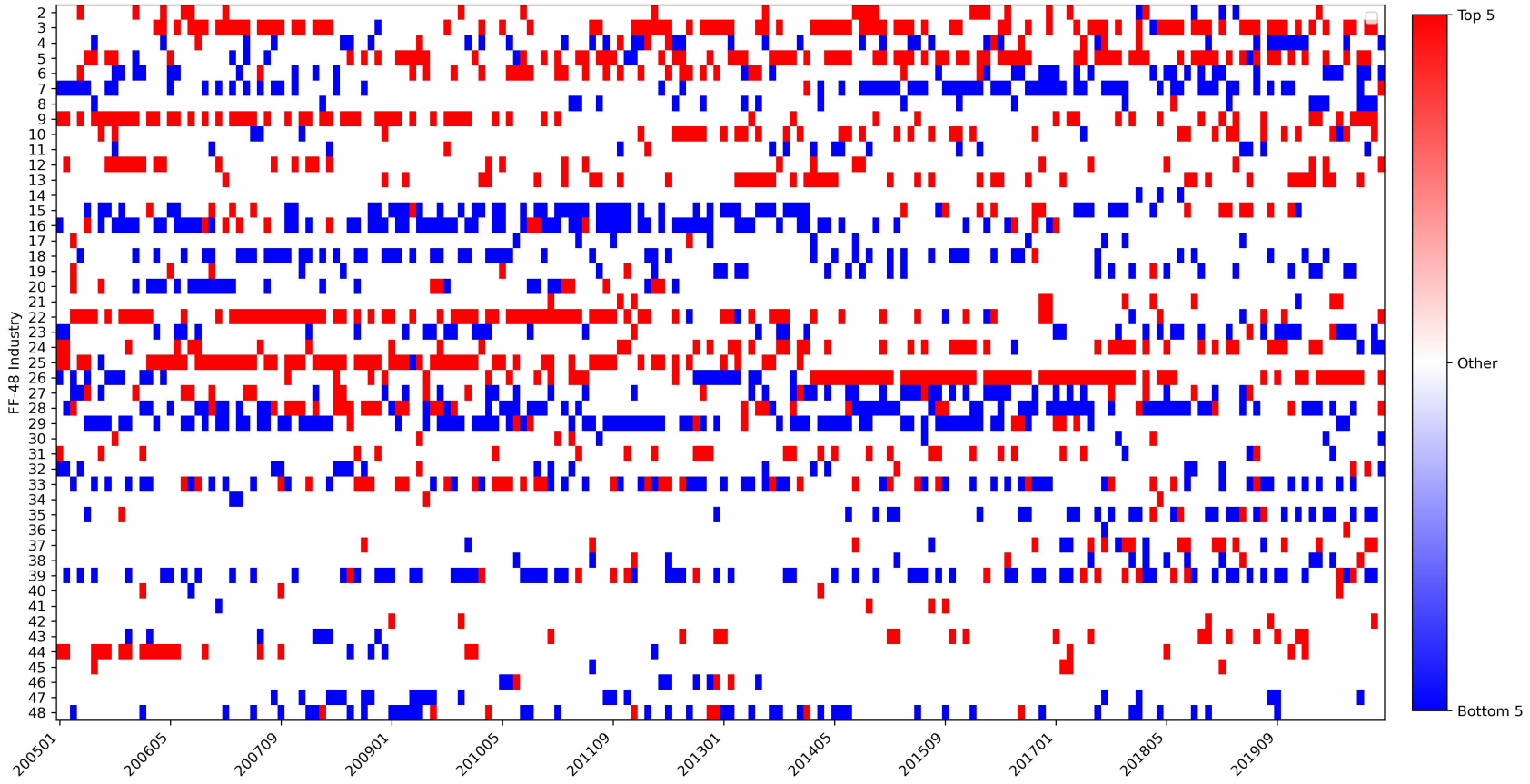


Figure 6. Time-Varying Network-Attributable Spread Changes Across Industries. This figure presents a heatmap of the time-varying network-attributable spread change (NSC) across industries. For each month and each Fama–French 48 industry, we compute NSC, defined as the average absolute difference between 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.

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, Francis A. Longstaff, Stephen Schaefer, and Ilya Strebulaev, 2011, Corporate bond default risk: a 150-year perspective, *Journal of Financial Economics* 102, 233–250.

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 Production Network Foundations and Idiosyncratic Volatility Spillovers **39**

A.1 Technology and First-Moment Production Network 39

A.2 From Revenue Propagation to Idiosyncratic Volatility 40

A.3 Interpretation and Economic Meaning 40

A.4 Link to Default Risk and Credit Spreads 41

B CDS Distribution Across Sectors **41**

C Firm Level Characteristics **42**

D Algorithm Details **43**

D.1 Graph Neural Networks 43

D.2 GNN-Attention 45

D.3 CNN 46

D.4 PCR and PLS 47

D.5 GBRT and RF 48

D.6 SVR 49

E Model Complexity and Stability **50**

F Robustness of GNN Under Various Specifications **51**

A Production Network Foundations and Idiosyncratic Volatility Spillovers

This section provides a structural microfoundation for the idiosyncratic volatility spillover network used in the main text. We begin with the explicit production technology and equilibrium mapping in Chen (2023), and then derive the second-moment spillover network $A = [a_{ij}]$ that governs default risk propagation.

A.1 Technology and First-Moment Production Network

Consider an economy with N firms indexed by $i = 1, \ldots, N$. Firm i produces output $Y_{i,t}$ using a Cobb–Douglas aggregator of intermediate inputs:

$$Y_{i,t} = X_{i,t} \prod_{j=1}^{N} \left(M_{ij,t}\right)^{(1-\alpha)\omega_{ij}},$$

where $X_{i,t}$ is firm-level productivity, $M_{ij,t}$ denotes the quantity of input j used by firm i, $\alpha \in (0,1)$ governs decreasing returns to intermediates, and $\omega_{ij} \geq 0$ is the input share of firm j in firm i's production. The input weights satisfy

$$\sum_{j=1}^{N} \omega_{ij} = 1 \quad \text{for all } i.$$

Productivity follows an idiosyncratic AR(1) process:

$$\log X_{i,t} = \rho_i \log X_{i,t-1} + \sigma_i \varepsilon_{i,t}, \qquad \varepsilon_{i,t} \sim \mathcal{N}(0,1) \text{ i.i.d.}$$

Let $\Omega \equiv [\omega_{ij}]$ denote the **first-moment production network**, capturing direct input-output linkages.

Under perfect competition and cost minimization, equilibrium prices and quantities imply a linear mapping from productivity shocks to firm-level revenues. Define the Leontief inverse

$$L \equiv (I - (1-\alpha)\Omega)^{-1} = \sum_{k=0}^{\infty} (1-\alpha)^k \Omega^k,$$

which captures infinite rounds of input-output feedback.

Let $z_t \equiv (\log X_{1,t}, \ldots, \log X_{N,t})^\top$. Then firm-level log revenues satisfy

$$\log R_{i,t} = \text{const}_i + \sum_{j=1}^{N} L_{ij} z_{j,t}, \tag{15}$$

where L_{ij} measures firm i's total (direct and indirect) exposure to productivity shocks originating at firm j.

A.2 From Revenue Propagation to Idiosyncratic Volatility

Equation (15) implies that firm i's revenue growth innovation can be written as

$$\Delta \log R_{i,t} = \sum_{j=1}^{N} L_{ij}\sigma_j \varepsilon_{j,t}.$$

The conditional variance of firm i's revenue growth is therefore

$$\mathrm{Var}_t\left(\Delta \log R_{i,t}\right) = \sum_{j=1}^{N} L_{ij}^2 \sigma_j^2.$$

This expression shows that idiosyncratic volatility at firm i is not firm-specific: it aggregates squared exposures to shocks originating at all upstream firms j.

To isolate the **relative contribution** of each upstream firm to firm i's idiosyncratic volatility, define the idiosyncratic volatility spillover weights

$$a_{ij} \equiv \frac{L_{ij}^2 \sigma_j^2}{\sum_{k=1}^{N} L_{ik}^2 \sigma_k^2}.$$

By construction,

$$a_{ij} \geq 0, \qquad \sum_{j=1}^{N} a_{ij} = 1 \quad \text{for all } i.$$

Let

$$A \equiv [a_{ij}]$$

denote the **idiosyncratic volatility spillover network**. In contrast to the production network Ω, which governs first-moment output propagation, the matrix A captures how idiosyncratic shocks contribute to second-moment (volatility) comovement across firms.

A.3 Interpretation and Economic Meaning

The spillover weight a_{ij} measures the fraction of firm i's idiosyncratic volatility attributable to shocks originating at firm j. A large a_{ij} indicates that firm j is an economically important upstream contributor to firm i's risk, either directly or through higher-order production chains.

Importantly, A is an endogenous transformation of the production network Ω:

- Ω governs how shocks propagate in levels;
- the Leontief inverse L aggregates infinite rounds of propagation;
- squaring L_{ij} maps first-moment exposure into second-moment risk;
- normalization delivers a row-stochastic volatility spillover matrix.

Thus, even when primitive productivity shocks $\varepsilon_{j,t}$ are purely idiosyncratic, the production network generates a rich structure of volatility spillovers summarized by A.

A.4 Link to Default Risk and Credit Spreads

Because firm i's asset volatility is increasing in $\sum_j L_{ij}^2 \sigma_j^2$, the spillover matrix A directly enters firm i's distance to default through the asset volatility channel. As shown in Section 2, larger downstream exposure a_{ij} lowers distance to default and increases default probabilities and CDS spreads.

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.

FIGURE 1 ABOUT HERE

The updating scheme is articulated as follows:

$$h_i^{(k+1)} = AGG\left\{ACT\left(DP\left(BN\left(\omega^{(k)}\hat{h}_i^{(k)} + b^{(k)}\right)\right)\right), j \in \mathcal{N}(i)\right\}, \tag{16}$$

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 (16) describes the rule for updating node embeddings within a GNN comprising a single hidden layer. The embeddings from neighboring nodes are aggregated, and the resultant 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 **??**.

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:

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

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 (17) , $a_{i,j}$ 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:

$$\hat{H}^{(k)} = \begin{cases} D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} H^{(k)} W^{(k)} & \text{if } A \text{ is symmetric} \\ D^{-1} \hat{A} H^{(k)} W^{(k)} & \text{if } A \text{ is asymmetric} \end{cases} \tag{18}$$

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 (18), 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'[15].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:

$$\begin{cases} \text{node attention}^{(k+1)} = DP\left(ACT\left(BN\left(\text{node attention}^{(k)}\right)\right)\right) \\ h_i^{(k+1)} = AGG_2\left\{AGG_1\left\{DP\left(ACT\left(BN\left(\omega^{(k)}\hat{h}_i^{(k)} + b^{(k)}\right)\right)\right), \text{Node attention}^{(k)}\right\}, j \in \mathcal{N}(i)\right\} \end{cases} \tag{19}$$

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 (19) 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 (16), 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:

$$\hat{H}^{(k)} = \begin{cases} \left(\left(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}\right) \odot \text{edge attention}^{(k)}\right) H^{(k)}W^{(k)} & \text{if } A \text{ is symmetric} \\ \left(\left(D^{-1}\hat{A}\right) \odot \text{edge attention}^{(k)}\right) H^{(k)}W^{(k)} & \text{if } A \text{ is asymmetric} \end{cases} \tag{20}$$

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

[15]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:

$$\text{edge attention}^{(k+1)} = DP\left(ACT\left(BN\left(\text{edge attention}^{(k)}\right)\right)\right), \tag{21}$$

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 (21) and (20), 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.

$$h_v^{(k+1)} = AGG\left\{ACT\left(DROPOUT\left(BN\left(\omega^{(k)} h_v^{(k)} + b^{(k)}\right)\right)\right), v = 1, 2, ..., N\right\} \tag{22}$$

The equation (22) 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. [16] The PCR model is represented as:

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

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

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

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

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

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

The number of principal components is determined through the validation set. PCR seeks linear

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

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.

Similar to PCR, PLS also seeks the p principal components Ω so that

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

and this number is tuned from the validation set.

However, its optimization goal is different:

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

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

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:

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

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. [17]

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 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

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

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 $112\times12 = 1,344$ 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.