Graph Machine Learning for Asset Pricing: Traversing the Supply Chain and Factor Zoo*

Agostino Capponi† J. Antonio Sidaoui‡ Jiacheng Zou§

This draft: January, 2025

Abstract

We propose a nonparametric method to aggregate rich firm characteristics over a large supply chain network to explain the cross-section of expected returns. Each target firm receives a nonlinearly constructed pricing signal passed from neighboring firms that are within d-hops on the supply chain network. Analyzing all US-listed stocks with supply chain data, our model achieves over 50% higher out-of-sample Sharpe ratios compared to models using only direct suppliers and consumers, outperforming Fama-French five-factor and principal component models. Through a graph-Monte Carlo experiment, we demonstrate the interplay between d and degree centrality, showing that the most central firms are twice as sensitive as peripheral firms. Our recommended $d = 6$ balances bias-variance and ensures robustness.

Keywords: Supply chain, cross-section of expected returns, machine learning, graph learning, interpretability of machine learning.
JEL classification: C33, C38, C45, C52, C55, G12

*We thank Garud Iyengar for thoughtful comments. We also thank the conference participants at the INFORMS Annual Meeting for helpful comments. We thank Dehan Cui for excellent research assistance.

†Columbia University, Department of Industrial Engineering and Operations Research, Email: ac3827@columbia.edu.

‡Columbia University, Department of Industrial Engineering and Operations Research, Email: j.sidaoui@columbia.edu.

§Corresponding author. Columbia University, Department of Industrial Engineering and Operations Research, Email: jz3865@columbia.edu.

1 Introduction

Modern economies are interconnected through intricate supply chains. As highlighted in Cohen and Frazzini (2008), investors are increasingly recognizing supply chains as a critical source of risk, given their role in capturing inter-firm business linkages and explaining the cross-section of expected stock returns.[1] Ding et al. (2021) demonstrate that the cross-section of stock returns varies not only with firm characteristics but also with their supply chains, using supply chain disruptions during the COVID-19 pandemic as a source of identification.

Despite evidence emphasizing the importance of understanding the risk premium associated with supply chain risks alongside other firm-level risks, a systematic asset pricing framework which integrates firm characteristics proposed in the "factor zoo" literature (Feng et al., 2020a) and supply chain relationship data remains absent. A key challenge lies in the fact that supply chains are inherently better represented as graph structures, while most asset pricing models rely on panel data comprising returns and firm characteristics.

We develop the first framework for systematic asset pricing with supply chains, making both methodological and empirical contributions. To the best of our knowledge, our framework is the first to directly incorporate firm-level supply chain data as conditioning information to estimate supply chain-based asset pricing models. Our approach is a novel machine learning (ML) framework that is capable of explaining returns using rich firm characteristics and higher-order relationships in supply chains. Extending state-of-the-art message-passing graph neural networks (GNNs), our approach estimates a flexible return estimation kernel that adaptively integrates firms' characteristics as it traverses supply chain graphs (SCGs). Moreover, as pointed out by Cochrane (2011), the challenge is to find "*which characteristics really provide independent information about average returns ... (vs) subsumed by others*". Therefore, we subsequently orthogonalize against prominent benchmark models that are also conditioning on the rich firm characteristics identified by Freyberger et al. (2020), and independently capture supply chain linkage risks that explain the expected returns.

Methodologically, supply chain data introduces multiple layers of complexity. Supply chain linkages are often missing from observed data, and their missingness often results from strategic decisions made by firms.[2] Moreover, supply chains evolve as firms adjust their partnerships. Such complications are also often not captured by data, or captured with lags and censoring. Existing

[1] Supply chain linkages introduce risks and vulnerabilities, such as shortages of essential inputs and surges in the prices of intermediate goods (Birge et al., 2023; Capponi et al., 2024). These vulnerabilities have been further amplified by rising geopolitical tensions, including the energy crisis triggered by the war in Ukraine (International Energy Agency, 2024), as well as the distress in supply chains experienced during the pandemic (Wall Street Journal, 2024). We refer to Babich and Birge (2021) for a comprehensive review, highlighting how operational resilience, risk management, and financial stability are interwoven in today's complex supply networks.

[2] Firms' disclosure behavior of supply chain data is an active field of research in finance, economics and management science. In the supply chain transparency literature, for instance, whenever the disclosures may build brand trust among consumers, firms may choose to disclose supply chain linkages (Mohan et al., 2020). See Sodhi and Tang (2019) for a detailed review of current discussion on transparency. From the financial economics perspective, firms may choose supply chain linkages for non-production reasons. For example, firms may strategically form supply chain relationships for tax avoidance (Cen et al., 2017), accompanied by reduced disclosures. In our research as is in many other similar work, we rely on supply chain data from government-imposed disclosures that are subject to firms' strategic disclosure choices.

asset pricing models lack the ability to incorporate these structural nuances and dynamic relationships. Last but not least, we should also take into account the rich literature of return anomalies (Freyberger et al., 2020; Hou et al., 2020) when modelling supply chain interconnectedness, making the problem very high-dimensional. Thus, we need to clear a primary hurdle of using supply chain data: how can we effectively leverage noisy, dynamic, and high-dimensional supply chain data to extract factors that are explanatory of the cross-section of average returns.

Our ML-based framework can utilize supply chain data for asset pricing, tackling the challenges articulated above. Firstly, high-dimensionality can be mitigated by our formulation of supply chains with firm characteristics as graphs with covariates. Each firm is a node, each linkage is an edge,[3] and for each month-firm, the node is associated with a return and a vector of firm characteristics. Modern GNNs are designed for such graphs, and have empirically showed great performance on much larger datasets such as the social network of Reddit (Hamilton et al., 2017) and protein molecules (Shi et al., 2021). As for the inherent missingness of supply chain data, our asset pricing model are pricing signals from observed supply chain linkages, and thus can be naturally interpreted as conditional information observed by investors. Moreover, due to the nature of our adaptive procedure, the dynamic nature of supply chain data *helps* our model to learn how partially removal or adding linkages impact our asset pricing factors.

Empirically, we show that a moderately complex GNN model with $d = 6$ (see Figure 3 for details)[4] captures higher-order relationships in the supply chain and yields 50% to 65% additional out-of-sample Sharpe ratios compared to Ridge and LASSO benchmarks that also leverage supply chain information.[5] This shows that GNNs are better able to capture the higher-order supply chain relationships with better asset pricing performance over linear models. Our proposed factors are tradeable assets that are orthogonalized against principal components (Giglio and Xiu, 2021; Lettau and Pelger, 2020) that summarize the firm characteristics. Thus, we extract signals from supply chains that make independent contributions to explaining the cross-section of average returns. Also, as we increase the model complexity to $d = 10$, the Sharpe ratio gains rise to 67% and 81% compared to Ridge and LASSO benchmarks.

We economically interpret the role of firm centrality in our neural network model in Figure 6 and Table 2 as we design a novel Monte Carlo-based experiment to assess how the model adapt to changes in the supply chain network. We find that the most central firms have on average 117% higher sensitivity than the most peripheral firms under our recommended GNN model. Addition-

[3]Full definition of the SCG's nodes and edges is given in §2.1. It is worth pointing out that we will be working on undirected and unweighted graphs to simplify the graph formulation, which sufficiently demonstrates how we adapt graph models for asset pricing. Extensions to more complex graphs such as hypergraphs, or sales-weighted graphs are interesting areas for future research.

[4]The parameter d of our GNN is interpretable as the radius of local sub-graph of the supply chain that our model is scanning. We discuss it in more details in §2.2.2.

[5]We construct Ridge and LASSO benchmarks to evaluate how these widely used linear models, which also incorporate supply chain relationship information, compare to our GNN models. Referred to as Neighboring Characteristics (NC) Ridge and LASSO models in our empirical section, these benchmarks do not have the full flexibility of GNN and only use nearest neighbor firms characteristics on the supply chain graph. See their definitions in §3.2.2, their firm-level return estimation performance in Table 1, and Sharpe ratios of their corresponding factors in Figure 3.

ally, the recommended model ($d = 6$) demonstrates average sensitivity 44% to 73% lower than other architectures, as the moderate model complexity can capture complex, multi-hop dependencies that are robust to noises induced by edge deletions, while offering a bias-variance trade-off. Overall, we find firm centrality and model complexity both play a role in determining the robustness of estimates, highlighting a balance between capturing relational complexity and maintaining stability in response to structural changes in the supply chain network.

Despite our focus on asset pricing, our proposed framework is broadly applicable to many economic problems with graph-panel *multimodal* learning. The term *multimodal* learning stems from the ML community (Ngiam et al., 2011), where data sources such as videos, photos and texts are in different *modes* and not naturally compatible, so researchers need to propose intelligently constructed models to jointly learn from different *modes* of data. In our case, we use financial economics-informed objective to combines two modes of data of SCG and firm-level returns and characteristics panel. Our framework generalizes to any problem of graph-panel bimodal data as long as the economic objective can be written as weighted sum of moment conditions. Economic problems with graphs are general and common, such as risk aggregation in networks (Acemoglu et al., 2012) and causal inference with interference graphs (Hudgens and Halloran, 2008), where the graphs are often also accompanied by rich covariates per node and per period. Since we make a step towards the conceptual question of graph-panel multimodal learning, the innovation of this paper is helpful for many other economic problems with graph and panel data.

The rest of the paper is organized as follows. Section 1.1 relates our work to the literature. Section 2 introduces the model and discusses the procedure to generate the asset pricing factors based on sorted portfolios. Section 3.1 discusses the data and Section 3.2 investigate the empirical findings. We dedicate Section 4 to interpret economic insights from the empirically estimated ML model. Section 5 concludes. Appendix A provides further details on the empirical analysis, while Appendix B offers an in-depth discussion of the Transformer architecture. To ensure reproducibility, our code is available on GitHub at https://github.com/agcappo/SupplyChainAssetPricing.

1.1 Related Literature

Our paper contributes to the growing body of literature on utilizing machine learning and big data to estimate latent factor models for cross-sectional asset pricing. In particular, the proliferation of predictive firm characteristics, often referred to as the "factor zoo", has motivated research focused on unifying these characteristics within cohesive and interpretable frameworks. Feng et al. (2020b) propose new statistical methods for selecting useful factors conditional on high-dimensional factors. Gu et al. (2020) compare the performance of various machine learning models to explore how non-linearities enhance signal extraction from characteristics, thereby improving return estimation. Chen et al. (2024) extend the application of machine learning beyond prediction by leveraging deep learning and moment conditions derived from no-arbitrage principles. Freyberger et al. (2020) develop nonparametric methods to select sparse models, highlighting the relevance of bias-variance tradeoff in the low signal-to-noise setting of finance. Farmer et al. (2023) shows that

return predictability is time- and firm-dependent.

Latent factor models by principal component analysis (PCA) are efficient ways to represent rich characteristics. The foundational work by Connor and Korajczyk (1988) directly links principal components to arbitrage pricing theory by a large cross-section of stocks. Panel-level explanatory power can be improved if additional preference on temporal vs cross-sectional explainability is specified, such as those proposed by Kozak et al. (2020); Lettau and Pelger (2020); Giglio and Xiu (2021); Bryzgalova et al. (2023). They offer alternative PCA objectives that mixes time-series and cross-sectional variations beyond the canonical form, and interpret economically motivated parameters that address the factor zoo's scope and variability. Kelly et al. (2019) presents a model that jointly estimates how to map firm characteristics to latent risk factors, and time-varying loading on the factors conditional on characteristic. Together, these studies underscore ML's potential in clarifying the factor structure within asset pricing, addressing both model parsimony and predictive power.

Herskovic (2018) examines how production networks and supply chain dynamics generate systematic risks that impact asset returns. Their theoretical model predicts that the graph centrality and sparsity of supply chain networks are key determinants of return spreads.[6] Their results rely on a structural model, and show that disruptions often lead to volatility clusters across firms within the same supply network. These studies highlight the importance of understanding supply chains as vehicles for risk propagation and underscores the need for models capable of capturing these interdependencies in asset pricing. Our approach is fundamentally different as we impose no structural model between firm returns and supply chain relationships. Instead, we propose a data-driven method to adaptively pick up the supply chain's non-linear influence as well as taking a large amount of firm-level information into account at the same time. Our sensitivity analysis further examines the impact of degree centrality on expected returns cross-sectionally.

Ding et al. (2021) reports the cross-section of stock returns reaction to COVID-19 cases in fact depend on characteristics. Specifically, pandemic-related return drops vary as they condition on firm characteristics such as firm size, firm finances (credit and profit levels) as well as supply chain placements. These findings underscore the role of supply chains as a systematic source of risk for asset returns. Our research leverages this finding, but shift the focus from exogenous shocks to the temporal variations in supply chain linkages, analyzed in conjunction with panels of firm characteristics.

A few studies have documented the significance of supply chain disruptions as predictors of financial performance. Wu (2024) proposes a text-based metric for assessing firms' exposure to supply chain risk, which captures how disruptions reverberate across firms with shared suppliers or customers. Agca et al. (2022) examine credit shock propagation within supply chains, showing that

[6]Some studies have analyzed the spillover effects of supply chain disruption. Carvalho et al. (2021) leverage the exogenous and regional nature of the East Japan earthquake shock to demonstrate the significant second-order effects of supply chain disruptions, which impact not only direct suppliers and customers of disaster-hit firms but also their indirect counterparts. Similarly, Barrot and Sauvagnat (2016) provide evidence of how idiosyncratic shocks propagate along supply chains, influencing the valuations of firms both directly and indirectly expoosed to the initial shock.

financial constraints in upstream firms can cascade down the supply chain, affecting downstream firms' stock returns. This finding supports the argument that understanding supply chain structures is necessary for asset pricing, as these relationships carry material risk information. Agrawal and Osadchiy (2024) extend this perspective by analyzing inventory productivity within supply chains, showing how firm-level operational data can predict stock returns. These contributions highlight the critical role of incorporating operational dependencies into asset pricing models to account for risk factors arising from supply chain linkages. Building on this foundation, we provide robust empirical evidence that supply chain information serves as a leading indicator for asset pricing. Specifically, we advance this stream of research by introducing a method that enables learning from any subset of the supply chain relationship network, without being constrained by pre-specified link directions or neighborhood structures.

The use of ML techniques in asset pricing has opened new avenues for exploring non-linear relationships and complex data structures that traditional models may overlook (Kelly et al., 2019; Gu et al., 2020; Chen et al., 2024). Our approach aligns with this trend, as we employ GNN to model the intricate web of supply chain relationships. By doing so, we contribute to the ML asset pricing literature by introducing a model that not only captures firm-specific information but also integrates relational and graph data to uncover latent factors in stock returns. This data-driven approach allows us to extract predictive signals from high-dimensional supply chain data, contributing to the "factor zoo" literature in asset pricing (Hou et al., 2020). Specifically, we use rich firm-level characteristics defined in Freyberger et al. (2020) as conditioning information for firms on the supply chain, and then compute the factors orthogonal to models computed by such firm characteristics so that we isolate the independent signals generated by the convolutions of SCG and firm information.

The studies by Chen et al. (2024) and Farmer et al. (2023) show that deep learning frameworks provide the necessary flexibility and scalability to model large datasets, which are essential when dealing with high-dimensional supply chain networks. By incorporating GNNs into our analysis, we contribute to this literature by adapting a model specifically designed to handle relational data, allowing for a nuanced understanding of firm interactions within supply chains.

In asset pricing with panel data, Kelly et al. (2022, 2024) find that complex models exhibit superior theoretical properties—partly due to the double-descent phenomenon in statistical learning (Hastie et al., 2022)—and deliver better empirical performance, including significant improvements in market timing Sharpe ratios. In contrast, our framework jointly learns from the graph data of supply chains and the panel data of returns and firm characteristics within a finite sample setting. A key feature of our method is a directly interpretable complexity parameter that defines the size of the local subgraph within the supply chain network. As this size increases, we derive asset pricing signals through more intricate convolutions on the supply chain network, resulting in improved performance. However, when the complexity parameter becomes too large, GNN performance deteriorates due to the well-documented "over-smoothing" phenomenon (Chen et al., 2020). Additionally, we observe that models with excessively high complexity parameters exhibit

greater variance. This underscores the importance of balancing complexity in graph-panel joint learning within a finite sample. While sufficient complexity is necessary to capture higher-order patterns, one consideration of the bias-variance tradeoff is essential to optimize performance, one should be mindful of the bias-variance tradeoff.

Notations: Unless otherwise specified, we use lower-case a to denote a scalar, bold lowercase $\mathbf{a}$ to denote a vector and bold uppercase $\mathbf{A}$ to denote a matrix. Firms are indexed by subscript i and time is indexed by subscript t. For an already defined firm-time varying variable a_{it}, we use $\mathbf{a}_t$ to denote the cross-sectional vector of the same variable at time t, and $\mathbf{a}_i$ to denote the time-series vector of the same variable for firm i.

2 Methodology

We introduce our approach by describing the detailed end-to-end procedure. First, we establish graph notation for supply chain networks to efficiently represent complex, high-dimensional information. Second, we discuss our design of a GNN that leverages the supply chain graph to non-linearly integrate firm characteristics through graph traversal. GNN embeddings are used for subsequent steps. Lastly, we discuss the use of PCA to decompose the embedding, projecting out firm characteristics to extract independent drivers of supply chain risks.

2.1 Supply Chains as Graphs in Asset Pricing

Our goal is to enable asset pricing with any combination of the supply chain relationship and firm characteristics data, which requires first developing a mathematical framework to jointly accommodate both. We proceed to introduce the necessary notations for firm characteristics data, and then discuss the methodology for integrating it with supply chain linkages. At time t, suppose that there are N_t firms in the universe of firms with observations of firm characteristics.[7] Let Y_{it} denote the excess return of the ith firm at time t, and $\mathbf{x}_{it}$ represent the associated vector of time-varying firm risk characteristics, such as those constructed in Freyberger et al. (2020). Both the characteristics and excess returns are observed with monthly frequencies, i.e. the subscript t encodes the observation month. We use p to denote the dimension of $\mathbf{x}_{it}$, and $\mathbf{X}_t \in \mathbb{R}^{N_t \times p}$ to denote the entire cross-section of firm characteristics at time t, concatenated row-wise as a matrix. The principal learning problem in asset pricing is a conditional estimation problem of returns using the risk characteristics with economics-informed objectives and constraints. Modern empirical asset pricing researchers expand this learning problem with ML techniques that allow for a non-linear functional form and a large p, a large N_t or both.

For supply chain asset pricing, we aim to describe the relationships of returns and firm characteristics conditioning on all supply chain information. Supply chain and asset pricing data at any

[7] We provide more details on the comprehensive universe of U.S. listed firms that we consider in the empirical section's §3.1.

time t is a very high-dimensional object if we express it naively in a vector form, even when p is not too large. We will soon revisit the high-dimensional nature of the data once we introduce the graph notations by contrasting (1) against (2). Supply chain data in our scope of research is observed on an annual basis. However, firms may have their observed supply chain relationship data updated at different months in the year, so the subscript t still encodes the observation month. We will revisit more details on the data observation frequencies in Section §3.1.

Graphs are the natural mathematical object to efficiently describe linkage data. We describe the entire supply chain network of *all firms* at any time t as a single graph.[8] In graph notation, we denote firm i as node i and the set of nodes available in the supply chain data at time t as V_t. Note that V_t represents a subset of the universe of firms, as there may be firms for which we have stock returns or firm characteristics but that are not included in the supply chain data, i.e., $|V_t| < N_t$. There is an edge e_{ijt} between nodes i and j whenever there is a direct supply chain linkage at time t between firms i and j. At time t, we have a set of edges E_t given by $E_t = \{e_{ijt} : i \in V_t, j \in V_t\}$. In addition, we denote the neighbors of node i at time t as $\mathcal{N}_t(i)$.

Supply chain graph (SCG) at time t is thus expressed as $SCG_t = (V_t, E_t)$. In our asset pricing problem, we associate the firm characteristics with its corresponding node, and only require a D_t^{Graph}-dimensional vector to jointly express the asset pricing and relationships data:

$$D_t^{\text{Graph}} = (p+1) \times |V_t| + |E_t|. \tag{1}$$

The first additive term in (1) is due to each firm having $(p+1)$-dimensional information of return and firm-level characteristics, multiplied by $|V_t|$ firms on the graph. The second additive term represents the relationships.

To see that the graph is an efficient way to represent the joint supply chain and firm characteristics information, we consider a naive alternative: for each pair of firm (i, j), we have returns and characteristics $(Y_{it}, \mathbf{x}_{it}, Y_{jt}, \mathbf{x}_{jt}) \in \mathbb{R}^{2(p+1)}$, while there are $|E_t|$ pairs. This adds up to a naive vector representation of all pairwise firm information as a D_t^{Naive}-dimensional vector of data:

$$D_t^{\text{Naive}} = 2(p+1) \times |E_t|. \tag{2}$$

Since each node in our graph has at least one edge by definition, $|E_t| > |V_t|$, and $D_t^{\text{Naive}} > D_t^{\text{Graph}}$ and the graph representation is more succinct. In the worst case, $|E_t|$ can be as high as $\binom{|V_t|}{2} = \frac{1}{2}|V_t|(|V_t|-1)$, so that the dimensional difference between D_t^{Naive} and D_t^{Graph} is on the order of $p \times |V_t|^2$.

Our next step is to leverage flexible ML methods to jointly extract information from $\mathbf{X}_t$ and SCG_t, aiming to explain the cross-section of average returns using our data representations.

[8] Locally, a pair of firms may appear in different supply chains, but in our graph representation of the entire supply chain network, we draw an edge between a pair of firms only once to indicate the existence of a direct linkage.

2.2 Pricing Kernel: Cross-Sectional Graph Neural Network

To capture the interactions within the supply chain, we implement a GNN model that propagates information through the graph with an algorithm known as *message passing*.[9] This section serves as a succinct description of the techniques that we use and are important for understanding how we achieve learning from both SCGs and the firm characteristics. At each layer l, the GNN updates the "knowledge" of each node i by aggregating information from its neighboring nodes $j \in \mathcal{N}_t(i)$ and combining this with its own features. Intuitively speaking, for each firm i the GNN computes a nonlinear weighted average of characteristics of firms in i's local subgraph, which can best explain firm i's return.

2.2.1 Message Passing

GNNs are distinct from feed-forward and recurrent neural networks previously explored in ML-based empirical asset pricing research (Gu et al., 2020; Chen et al., 2024), as they are specifically designed to meet the unique requirements of graph-based learning. We consider GNNs that consist of d layers, with each layer iteratively passing messages between nodes according to the graph's structure. Mathematically, at time t we can describe the computation at layer l as follows:

1. **Message Calculation:** For each node i, we compute a message $m_{ij}^{(l)}$ from each neighbor j in its local neighborhood $\mathcal{N}_t(i)$. The message $m_{ij}^{(l)}$ is typically a function of both the feature vector $h_j^{(l-1)}$ of the neighboring node j and the feature vector $h_i^{(l-1)}$ of node i itself from the previous layer:

$$m_{ij}^{(l)} = f_{\text{msg}}(h_i^{(l-1)}, h_j^{(l-1)}, \mathbf{x}_{it}, \mathbf{x}_{jt}), \tag{3}$$

   where f_{msg} is a differentiable function that may also depend on characteristics $(\mathbf{x}_{it}, \mathbf{x}_{jt})$, capturing their supply relationship conditioning on the characteristics of both firms.

2. **Aggregation:** In the aggregation step, node i aggregates all incoming messages from its neighbors to form a single, summarized message. The aggregation function AGG is the widely used notation in the computer science graph learning literature, and must be a permutation-invariant function (such as summation, mean, or max) that consolidates messages in a functional form that does not depend on how many firms are in $\mathcal{N}_t(i)$:[10]

$$m_i^{(l)} = \text{AGG}(\{m_{ij}^{(l)} : j \in \mathcal{N}_t(i)\}). \tag{4}$$

   This aggregation captures the influence of all neighboring nodes on node i, distilling their collective information into a single vector.

[9] GNNs are a rapidly evolving field, with a remarkable volume of new research emerging daily. For a comprehensive introduction to the GNN framework, see Hamilton (2020).

[10] This permutation and size invariant requirement rules out many functions such as linear weighted sums, which would require the weighting matrix to change depend on cardinality of $\mathcal{N}_t(i)$.

3. **Update:** The node feature $h_i^{(l)}$ at layer l for $l = 1, 2, ..., d$ is then updated by combining its aggregated message $m_i^{(l)}$ with its current feature vector $h_i^{(l-1)}$. This combination is performed using a transformation function f_{upd}, which could be a simple weighted summation followed by a non-linear activation:

$$h_i^{(l)} = f_{\text{upd}}(h_i^{(l-1)}, m_i^{(l)}). \tag{5}$$

Here, f_{upd} serves as a learnable function that adapts to optimize the model's accuracy. A typical choice for f_{upd} is a neural network layer that applies weights and biases to linearly transform $h_i^{(l-1)}$ and $m_i^{(l)}$, followed by an activation function.

Lastly, we collect the output of message passing (i.e. the last hidden layer's neurons) $h_i^{(d)}$ as the embeddings. The output layer is a linear regression of returns Y_{it} on the embeddings.

The specific choices of $(f_{msg}, \text{AGG}, f_{upd})$ will impact empirical performances (Kipf and Welling, 2016; Hamilton et al., 2017; Shi et al., 2021; Brody et al., 2021), and their trade-offs contain many idiomatic technical motifs. We use the well-known transformer architecture, and defer its details to Appendix B. In our empirical analysis, as exemplified in Table 1, we explicitly specify the model as TransformerGNN. However, for brevity, we refer to it as GNN throughout the rest of the paper. To formalize the GNN's learning process, we define an optimization objective that minimizes the error between estimated and actual firm-level returns, enabling the model to effectively optimize its parameters. Let Θ denote the set of all model parameters, including weights and biases in the GNN layers and other parameters such as attention weights in the multi-head Transformer mechanism that we use (see in Appendix B's (22) for f_{msg}, (23) for AGG and (24) for f_{upd}). The objective is to minimize the following empirical risk function over the training dataset:

$$\min_{\Theta} \mathbb{E}_{(i,t)\sim\mathcal{D}_{\text{train}}} \left[(Y_{it} - g_i(\Theta; \mathbf{X}_t, SCG_t))^2 \right], \tag{6}$$

where:

- Y_{it} is the return for firm i at time t,
- $g_i(\Theta; \mathbf{X}_t, SCG_t)$ is the model's estimation function,[11] which is a function of the entire cross-section of firm characteristics $\mathbf{X}_t$ and the structure of the supply chain graph SCG_t,
- $\mathcal{D}_{\text{train}}$ is the training dataset comprising observations of firms over time.

In our empirical estimation, this minimization is achieved with Adam, which is a gradient-based stochastic optimization algorithm (Kingma and Ba, 2015) that iteratively adjusts the parameters in Θ to reduce the mean squared error (MSE) on the training set. The optimization process adjusts

[11] We denote the function g_i to highlight the networks are heterogeneous across i's, allowing the model to treat different firms' returns differently when given the same cross-sectional $(\mathbf{X}_t, SCG_t)$. However, the GNN does not need to be trained separately for each firm. Instead, the model is trained only once by rolling out the message passing of SCG_t with firm i as the target node, thereby presenting the graph-characteristics joint data differently across firms. Intuitively, the g_i's differ from each other by their vantage point of the SCG.

the GNN's parameters to align the non-linear combination of firm characteristics with returns, effectively capturing the joint impact of supply chain relationships and firm-specific attributes in explaining average returns.

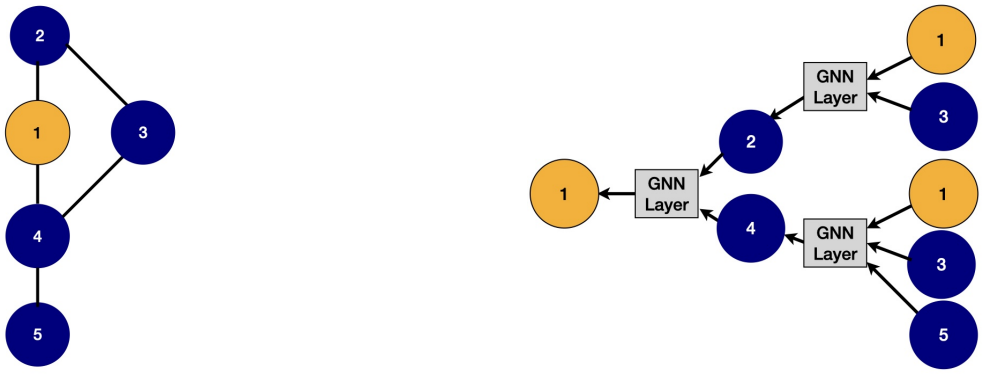
Finally, we emphasize that in equation (6), our supply chain-based pricing error can be replaced with other economic objectives, enabling the utilization of panel data comprising unit-time responses and covariates, along with a time-varying graph of units.

2.2.2 Layer Depth and Subgraph Size

The depth of the GNN, represented by the number of layers d, determines the size of each node's local subgraph that the model incorporates when updating node features. Each additional layer in the GNN allows information to propagate further across the graph, thereby increasing the number of nodes whose information influences a given target node. With d layers, each node receives messages from all nodes within d-hops (number of edges required to traverse) in the graph, effectively expanding its scope of information aggregation.

Visually, we provide the following diagrams to show how the message passing process works and highlight the connection between layer depth and subgraph size:

Figure 1: Message passing



(a) Example graph

(b) $d = 2$ message passing

In a message-passing GNN, each node's representation is iteratively updated by aggregating information from its neighbors. The number of GNN layers determines the number of "hops" each node's representation can access, effectively defining the size of the local subgraph that each node captures. In Figure 1, we focus on updating the features of the orange-colored node 1. Subfigure (a) illustrates a simple graph where node 1, the target node, is connected—either directly or indirectly—to nodes 2, 3, 4, and 5, which collectively form the local subgraph influencing node 1's representation. Subfigure (b) uses arrows to indicate the directions of message passing within this subgraph. For this example, we demonstrate the GNN's message passing process across two layers ($d = 2$). In the first layer, node 1 aggregates information from its immediate (one-hop) neighbors, nodes 2 and 4, updating its representation based on directly connected nodes. In the second layer,

node 1 can now receive information from nodes two hops away—nodes 3 and 5—via nodes 2 and 4. This expanded reach allows node 1's representation to reflect a broader local subgraph.

We also want to highlight that the message passed to the target node includes not only the characteristics of its neighbors but also those of the target node itself. Specifically in Figure 1, node 1, as the target node, receives messages from its immediate neighbors (nodes 2 and 4), which, in turn, embed node 1's own characteristics during the first layer of message passing. This recursive structure means that a regression of the target node on its local context—a standard approach in traditional asset pricing—emerges naturally as a special case of the message-passing mechanism in GNNs when $d \geq 2$. The ability of the GNN to incorporate the target node's features within the context of its broader supply chain relationships highlights the flexibility and generality of this framework, bridging conventional regression approaches and modern graph-based learning techniques. This property is central to the GNN's ability to capture both direct and higher-order dependencies within the supply chain network.

The GNN's layer depth controls how far each node can collect information within the graph. Multiple layers extend this to nodes further away, expanding each node's receptive field within the graph. Thus, the number of layers defines the size of the local subgraph influencing each node. More layers allow deeper neighborhood capture but can introduce noise if the extended neighborhood becomes less relevant to the target node.

In the context of a SCG, the layer depth d provides an interpretable metric for understanding how information is aggregated across supplier-customer relationships:

- **Small d (Local Subgraphs):** A shallow GNN with a small value such as $d = 1$ captures only immediate supply chain dependencies, typically limited to a firm's direct suppliers and customers. This limited scope may omit broader structural effects, as the model is constrained to first-order neighbors. From the firm operations perspective, a shallow network with small d focuses on immediate business linkages, which may capture only local risks and direct impacts on firm performance.

- **Large d (Expanded Subgraphs):** As d increases, each node's receptive field includes a wider portion of the graph, including information from second-, third-order and even higher-order suppliers and customers. This allows the model to account for indirect effects, such as the influence of a supplier's suppliers or a customer's customers on the target firm. For instance, if a supplier's own upstream partner experiences a disruption, this may eventually impact the firm in question. Therefore, larger values of d enable the GNN to capture more comprehensive supply chain effects, potentially revealing how upstream or downstream disturbances propagate through the network.

- **Bias-Variance Trade-off with Layer Depth:** While a greater layer depth d can theoretically improve model expressiveness by capturing more distant relationships, it introduces a trade-off between model bias and variance. Large d values can dilute the influence of closer, more relevant neighbors with noise from distant, less relevant nodes—a phenomenon

commonly referred to as the "information dilution" effect. This trade-off is particularly pronounced in sparse or dynamic supply chain networks, where distant nodes may introduce uninformative or even misleading signals. Additionally, excessive depth may lead to the well-known *vanishing gradient* and *oversmoothing* issue (Chen et al., 2020) in GNN, complicating model training and degrading estimation performance in out-of-sample tests.

Thus, determining the optimal layer depth d is a crucial part of model design. For our asset pricing task, we empirically identify an intermediate depth that balances capturing meaningful indirect supply chain relationships while minimizing the introduction of excessive noise. This depth aligns with the scope of influence that firms typically observe in practical supply chain contexts, where both local and moderately extended relationships play essential roles in risk transmission and performance prediction.

In summary, the layer depth d is a crucial hyperparameter that governs the model's sensitivity to the supply chain structure, with its interpretability rooted in the extent of indirect relationships it captures within the network.

2.2.3 Output Embeddings $\tilde{\mathbf{x}}_{it}$

After the final layer d of the GNN, each node i at time t is represented by an embedding vector $\tilde{\mathbf{x}}_{it} = h_i^{(d)} \in \mathbb{R}^{p'}$, where $h_i^{(d)}$ is the output feature vector from the last layer and p' is its dimension. Note that technically, the embedding dimension p' is not necessarily smaller or larger than the original firm characteristics dimension p, but typically we choose $p' < p$ to represent a nonlinear dimension reduction from the original feature space to the embedding space. This embedding $\tilde{\mathbf{x}}_{it}$ captures the combined information from the firm-specific features of node i and relational features derived from the message-passing process across the SCG.

The embeddings $\tilde{\mathbf{x}}_{it}$ serve as a compact representation of each firm's supply chain context, summarizing both its direct relationships and the broader network structure up to d hops away. These embeddings will later be used as inputs for our supply chain factors. However, it is important to emphasize that the GNN model functions more like a cross-sectional non-parametric kernel, combining firm characteristics based on the geometry of the supply graph. It does not yet directly capture temporal variations in returns.

2.3 Orthogonal Transformation by Principal Components

We inject temporal information by computing the principal component directions of the embeddings within the training set. Our embeddings $\tilde{\mathbf{x}}_{it} \in \mathbb{R}^{p'}$ capture high-dimensional features reflecting intricate cross-sectional dependencies within the supply chain. Our goal is to map these embeddings to directions that effectively explain time-series variations.

We are also interested in the performance of a linear factor model based on our factors, which constitutes a secondary but still crucial motivation for the PCA step. Note that $\tilde{\mathbf{x}}_{it}$ are by-products of neural networks, which have no inherent constraint on their statistical properties. In particular,

the directions of $\tilde{\mathbf{x}}_{it}$ may be correlated. Therefore, to identify *independent* new risk factors, as proposed by Cochrane (2011), we map the data to latent principal directions, allowing us to extract more robust insights and test whether we have uncovered any independent sources of explained average returns.

We compute the principal components by solving the following optimization problem:

$$\max_{\mathbf{V}} \quad \sum_{(i,t)\in\mathcal{D}_{\text{train}}} \text{Tr}(\mathbf{V}^\top \tilde{\mathbf{x}}_{it}^\top \tilde{\mathbf{x}}_{it} \mathbf{V}) \quad \text{s.t.} \quad \mathbf{V}^\top \mathbf{V} = \mathbf{I}_{p'}. \tag{7}$$

The matrix $\mathbf{V} \in \mathbb{R}^{p' \times p'}$ is the transformation matrix. The constraint of $\mathbf{V}$ being orthonormal ensures that the problem is uniquely identified. Let the k-th column be denoted as $\mathbf{v}_k$, representing the direction of the k-th principal component, sorted by their contributions to the variance of $\tilde{\mathbf{x}}_{it}$ during the training period. Then, for each principal component k at time t we have a firm-specific ϕ_{itk}:

$$\phi_{itk} = \tilde{\mathbf{x}}_{it}^\top \mathbf{v}_k, \quad k = 1, 2, ..., p'. \tag{8}$$

We denote the top K principal components as $\boldsymbol{\phi}_{it} \in \mathbb{R}^K$, which represent firm-level independent characteristics combining supply chain and firm attributes. That is, although up to as many as p' (the dimensions of the embedding) principal dimensions are available, we truncate and only use the most important K dimensions among them. These components are thus constructed from a two-step process. First, we apply a nonlinear, cross-sectional transformation using the GNN, which captures relationships between supply chain and firm-specific data. Second, we perform a linear transformation via PCA to extract the dominant dimensions of variation.

We interpret the leading components of $\boldsymbol{\phi}_{it}$ as the transformed embedding dimensions that explain the largest share of variation in the data, which are later used to construct our asset pricing factors. Our approach follows a logically nested framework, ensuring that the discussion of the number of factors aligns naturally with their economic and statistical ordering. The construction, therefore, provides us with a coherent foundation for analyzing the asset pricing significance of the proposed supply chain-based factors.

2.4 Asset Pricing Factors

We now create asset pricing factors from the proposed principal components $\boldsymbol{\phi}_{it}$. Interpreting $\boldsymbol{\phi}_{it}$ as firm-month supply chain characteristics, we first sort firms into deciles based on these characteristics following Fama and French (1992), and then create portfolios based on deciles to represent tradable assets based on $\boldsymbol{\phi}_{it}$. Specifically, each firm is assigned to one of 10 possible bins based on its ϕ_{itk} value along the k-th principal direction. Then, we build a zero-cost long-short portfolio for each k. Specifically, for each principal component k and time t[12]:

[12]This sorted portfolio construction approach is the well-established method to connect machine learning firm-level estimation to the cross-section of expected returns in previous empirical asset pricing literature (Gu et al., 2020; Hou et al., 2020; Chen et al., 2024).

1. Sort all firms by their score ϕ_{itk}.

2. Define the top decile (highest scores) as the "long" portfolio and the bottom decile (lowest scores) as the "short" portfolio. Within each portfolio, invest equally into each firm.

3. Construct a zero-cost long-short portfolio by taking a long position in the top decile and a short position in the bottom decile and use the return as the value of F_{tk}^{LS}.

Each individual component F_{tk}^{LS} is tradeable, and we can naturally consider the number of factors as a varying parameter for our comparative analysis, as F_{tk}^{LS} is derived from the kth principal component of $\tilde{\mathbf{x}}_{it}$.

The only additional step is to guarantee the independent explainability of our supply chain asset pricing factors. We regress each long-short portfolio F_{tk}^{LS} on a set of benchmark factors, such as the leading principal components estimated from the portfolios of sorted firm characteristics (Giglio and Xiu, 2021; Lettau and Pelger, 2020). This orthogonalization step removes any overlap with traditional asset pricing factors, ensuring that the GNN factors represent new sources of return variation.

The regression model for each factor F_{kt} can be expressed as:

$$F_{tk}^{LS} = \eta_k^\top \mathbf{z}_t + \varepsilon_{tk}, \tag{9}$$

where $\mathbf{z}_t$ are the excess returns of the benchmark factors at time t, which are assumed to also correspond to the excess returns of zero-cost tradable assets, and ε_{tk} are the unexplained returns.

The η_k coefficients are estimated using OLS, and the resulting residuals from this regression represent the orthogonalized factors, which are uncorrelated with traditional asset pricing benchmarks. These residualized factors are thus novel latent factors derived from the GNN's analysis of supply chain networks and can be interpreted as the unique contribution of supply chain relationships to firm returns. Therefore, we use them as our asset pricing factors: [13]

$$F_{tk} = F_{tk}^{LS} - \hat{\eta}_k^\top \mathbf{z}_t. \tag{10}$$

The orthogonalized factors F_{tk} reflect firm-specific risk and return variations linked to the structure of supply chain relationships rather than either the FF5 or the leading RP-PC. By regressing out these factors, we can test whether supply chains combined with firm characteristics contain unique signals for asset pricing, capturing aspects of firm interdependence that are not represented in conventional factor models. This novel perspective allows for a more comprehensive understanding of the drivers of firm returns, particularly in interconnected industries where supply chain relationships play a crucial role.

Since our factors $\mathbf{F}_t^{LS}$ are zero-cost tradeable assets and $\mathbf{z}_t$ are also zero-cost tradeable assets, our residual $\mathbf{F}_t$ are also zero-cost tradeable assets that represent risk premium of the supply chain

[13] None of our factors are strongly correlated with the FF3 factors. For the first five of our factors, we find weak positive or negative correlations with the Book-to-Market, Size, and Market factors. See Figure 5.

factors.

Under the no-arbitrage condition, a stochastic discount factor (SDF) m_{t+1} exists, ensuring that the expected value of the discounted excess returns $\mathbf{r}_t$ of test assets is zero in expectation:[14]

$$\mathbb{E}_t[m_{t+1}\mathbf{r}_{t+1}] = 0. \tag{11}$$

By the definition of covariances, the SDF is linked to the test assets through the following moment conditions:

$$\mathrm{Cov}_t(\mathbf{r}_{t+1}, m_{t+1}) = \underbrace{\mathbb{E}[m_{t+1}\mathbf{r}_{t+1}]}_{=0} - \mathbb{E}_t[m_{t+1}]\mathbb{E}_t[\mathbf{r}_{t+1}]. \tag{12}$$

To arrive at the normalized covariance test for asset returns with SDF on one side and the risk premium on the other, we divide through by the variance of SDF to calculate:

$$\frac{\mathrm{Cov}_t(\mathbf{r}_{t+1}, m_{t+1})}{\mathrm{Var}_t(m_{t+1})} = -\mathbb{E}_t[\mathbf{r}_{t+1}] \cdot \frac{\mathbb{E}_t[m_{t+1}]}{\mathrm{Var}_t(m_{t+1})}. \tag{13}$$

Ross (1976) introduced the Arbitrage Pricing Theory (APT), which examines the scenario where m_{t+1} is modeled as a linear function of a set of factors. Under APT, the expected excess returns are explained by loadings on risk factors times the risk premium of the factors. We linearly expand the SDF in the special case of our supply chain factors, where the expected value of m_{t+1} is linear in both $\mathbf{F}_t$ and $\bar{\mathbf{F}}_t$ that are also excess returns – the $\mathbf{F}_t$ are our supply chain asset pricing factors and $\bar{\mathbf{F}}_t$ are other factors that are orthogonal to $\mathbf{F}_t$.

Since the factors are constructed as excess returns, we can reorganize equation (13) to isolate the expected excess returns on the left-hand side: [15]

$$\mathbb{E}_t[\mathbf{r}_{t+1}] = \Lambda_t \mathbb{E}[\mathbf{F}_t] + \tilde{\Lambda}_t^C \mathbb{E}[\bar{\mathbf{F}}_t]. \tag{14}$$

The regression (14) serves as the basis on which we empirically estimate and report the statistical significance of loadings (Λ_t) for our supply chain asset factors. The empirical results are discussed in Section §4 and visualized in Figure 4, where we observe many of the loadings are significantly non-zero.

In practice, the true non-supply chain factor space $\{\bar{\mathbf{F}}_t\}_t$ is unobservable, and we must rely on approximations $\mathbf{z}_t$ derived from the universe of proposed "factor zoo" models. By regressing out $\mathbf{z}_t$ in (10), our constructed factors $\mathbf{F}_t$ are orthogonal to $\bar{\mathbf{F}}_t$ only if $\bar{\mathbf{F}}_t$ is fully spanned by $\mathbf{z}_t$. However, if $\mathbf{z}_t$ does not fully span $\bar{\mathbf{F}}_t$, residual correlations between $\mathbf{F}_t$ and $\bar{\mathbf{F}}_t$ may persist. Importantly, this limitation pertains to the quality of the benchmark models $\mathbf{z}_t$ rather than the validity of the constructed factors $\mathbf{F}_t$ themselves. As long as $\{\mathbf{z}_t\}_t$ are improved to cover a larger region of the

[14] The standard derivation with prices and returns can be found in Cochrane (2009, Chapter 6), as opposed to our discussion that focuses on excess returns.

[15] We are simplifying the algebraic mechanics ude to the scope of our paper. The detailed steps of derivation for this step is part of the "Factor Models and Discount Factors" in §6.3 of Cochrane (2009).

column space $\{\bar{\mathbf{F}}_t\}_t$, the constructed factors $\mathbf{F}_t$ enjoy better approximated orthogonality with the non-supply chain factors $\bar{\mathbf{F}}_t$, thereby yielding an improved estimation of the loading Λ_t. We now have an empirical test of whether the expected returns have non-zero exposures to our supply asset pricing factors $\mathbf{F}_t$ by examining the loadings Λ_t in regressions (14), with the clear caveat that it is conditional on a given set of benchmark models that we control for.

2.5 Summary of the Procedure

To summarize, our proposed supply chain asset pricing factor estimation procedure consists of two stages: the first stage is on the firm level, and the second stage is on the factor level. In each stage, there are two steps respectively, so there are four steps in total:

Supply chain asset pricing model estimation and factor construction

---

Firm level stage: machine learning on the supply chain and firm characteristics

Step 1. Estimate graph neural network (GNN) model by optimizing (6).
Output: $\tilde{\mathbf{x}}_{it}$ that cross-sectionally combine firm characteristics and the supply chain relationships to explain firm returns; the estimated GNN model.

Step 2. Apply PCA on $\tilde{\mathbf{x}}_{it}$'s based on (7).
Output: $\boldsymbol{\phi}_{it}$ that are orthogonal and ordered by their PC rank, spanning the leading K eigenvectors of the eigenspace of the second moment matrix of $\tilde{\mathbf{x}}_{it}$.

Factor level stage: sorting and projections to construct factors

Step 3. Construct long-short portfolios from sorting on each of $\boldsymbol{\phi}_{it}$'s K entries.
Output: $\mathbf{F}_t^{LS}$ are tradeable zero-cost assets.

Step 4. Compute asset pricing factor by regressing out benchmark model in (10).
Output: $\mathbf{F}_t$ are asset pricing factors orthogonal from benchmark.

---

3 Empirical Supply Chain Asset Pricing

We apply our methodology to investigate how supply chain data and firm characteristics explain expected returns. We present the sources and summary statistics of our data, including firm characteristics and supply chain relationships. In our empirical analysis, we begin by comparing the firm-level return estimation performance of the GNN model against benchmark models. Finally, we analyze the Sharpe ratios of portfolios formed using supply chain asset pricing factors.

3.1 Data

We combine firm-level financial data with supply chain relationship data. Our dataset captures both firm-specific and relational information essential for understanding the impact of supply chain structures on asset pricing.

3.1.1 Firm-Level "Factor Zoo" Data

We compute monthly stock excess returns Y_{it} and lagged firm characteristics $\mathbf{x}_{it}$ from publicly listed firms on the NYSE, Nasdaq, and NYSE American (formerly known as the American Stock Exchange). The time period is from January 1977 to December 2023, providing a longitudinal view of firm performance across varying market conditions. The firm characteristics $\mathbf{x}_{it}$ include $p = 64$ lagged financial indicators defined by Freyberger et al. (2020). The firm characteristics cover 6 categories of past return, investment, profitability, intangibles, value, and trading frictions, which overlaps with other well-known studies on "factor zoo" (Kozak et al., 2020; Hou et al., 2020). [16] Following the literature's practice (Kelly et al., 2019; Freyberger et al., 2020), we normalize $\mathbf{x}_{it}$ using the cross-sectional rank divided by the total number of firms N_t at time t as our $\mathbf{x}_{it}$ instead of directly using the values of the financial indicators. This cross-sectional rank normalization ensures that $\mathbf{x}_{it}$ falls within the bounded range of $[0, 1]$.

The returns Y_{it} represent the realized monthly excess return for firm i at time t, while the characteristics $\mathbf{x}_{it}$ provide a snapshot of the firm's financial health and performance. The combination of Y_{it} and $\mathbf{x}_{it}$ allows us to study the role of supply chain relationships in explaining variations in firm returns, beyond what traditional financial characteristics alone can capture. We use K. French's website to retrieve the risk-free rate and subtract it from the return in order to compute excess return. We source Y_{it} and $\mathbf{x}_{it}$ data from the Center for Research in Security Prices (CRSP) and Compustat Fundamentals databases on Wharton Research Data Services (WRDS).

It is worth pointing out especially that many of our $\mathbf{x}_{it}$ are not originally defined at a monthly frequency, as many accounting characteristics are disclosed on a quarterly or annual basis. Therefore, we lag them using common practice in the literature (Gu et al., 2020) to ensure there is no look-ahead bias: we assume that monthly characteristics are delayed by at most 1 month, quarterly with at least 4 months lag, and annually with at least 6 months lag. Therefore, for the learning task of explaining excess returns at month t, we use the most recent monthly characteristics at the end of month $t-1$, the most recent quarterly data by the end of month $t-5$, and the most recent annual data by the end of month $t-7$.

3.1.2 Supply Chain Relationship Data

To represent supply chain relationships, we use the supply chain graph SCG_t, where each node corresponds to a firm and each edge represents a supply chain linkage between firms. The relationship data is derived from mandatory disclosures under the SEC's SFAS 131 reporting standard,

[16] See more details in Freyberger et al. (2020)'s Table 1 and Internet Appendix.

which requires firms to disclose key customers that account for more than 10% of their sales. This results in a graph for each year, where the presence of an edge between node i and node j indicates that firm j and firm i's business relationship satisfies the SFAS 131 threshold.[17] We retrieve SCG_t from the Compustat Segment database on WRDS (Cohen and Frazzini, 2008; Cen et al., 2017), with data that begins in 1977 and ends in 2023. In total, there are 2521 unique firms present in our constructed SCG, which have both supply chain relationships and firm characteristics data. Firms for which there was no supply chain relationship data available were not included in the analysis.

We lag data when incorporating supply chain data. While many firm characteristics data is available at a monthly or quarterly frequency, supply chain disclosures are annual. We treat this temporal misalignment by holding each year's supply chain relationships constant across all months within that year, thereby allowing us to integrate the supply chain data into the monthly modeling framework. The lagging rule for supply chain data is the same as other annual financial data in §3.1.1: to explain returns at month t, we use the most recent annually disclosed supply chain data by the end of month $t-7$. This lagging treatment is why we consider the supply chain information as leading signals for our asset pricing question.

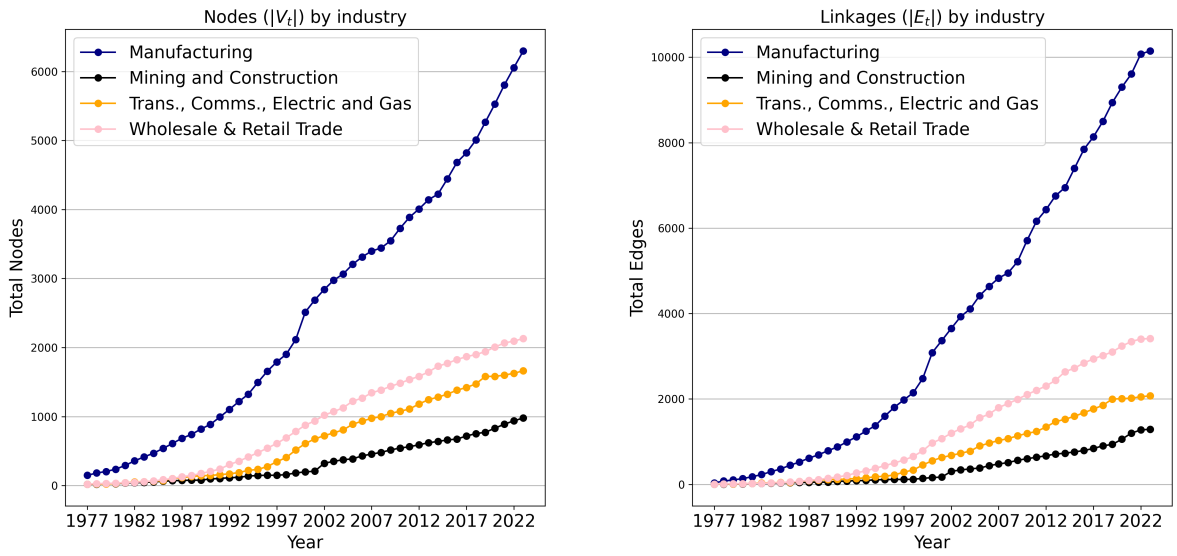
To effectively capture higher-order relationships within the supply chain, we retain firms that are not listed on the NYSE, NASDAQ, or NYSE American exchanges in our dataset. Including these unlisted firms allows the model to traverse multi-hop paths, which generates pricing messages passed between pairs of listed firms through their shared connections with unlisted intermediaries. This approach ensures that our model fully leverages the relational structure of the supply chain, capturing indirect dependencies that may impact publicly traded firms.

From the summary statistics of the SCG data, illustrated in Figure 2, we see temporal growth, sparsity, and cross-sectional differences among industries. Such patterns highlight important structural features of the supply chain network that influence model design and analysis. We also provide more granular summary statistics for the two leading industries of Manufacturing and Wholesale & Retail Trade in Table A.1, where we report the number of firms, supply chain links and average degree for each year in our observations.

First, the SCGs exhibit a clear upward trend in both the number of nodes (firms) and edges (supply chain relationships) over time, as shown in Figure 2. This temporal growth reflects the evolving complexity of the supply chain network, as more firms establish connections with suppliers and customers. To balance capturing long-term patterns in supply chain relationships with robust out-of-sample testing, we partition the data into a training period from 1977 to 2006 and a test period from 2007 to 2023. This break-point, as observed from Figure 2, provides us with a train and test set that roughly have the same number of supply chain relationships in each. The temporal growth of the SCG provides the model with progressively richer structural information, mirroring

[17] While SFAS 131 disclosure enables the construction of directed relationships (e.g., a directed edge from a supplier to a customer), we simplify the model by focusing on an undirected graph representation derived from the data. Additionally, although some edges include weights—such as sales percentages—we adopt an unweighted graph framework for tractability. Extending the model to incorporate directed or weighted graphs is a promising avenue for future research.

Figure 2: The number of nodes and edges by year



This figure depicts the number of firms $|V_t|$ in a given industry, and the total number of edges $|E_t|$ at time t connected to firms in a given industry in our observed supply chain graph data. The data is annual frequency. We identify a firm's industry by its Standard Industrial Classification (SIC) code.

the broader trend of economic integration over the past several decades. This expanded network structure offers the GNN more relational data for learning firm interdependencies, which is essential for accurately modeling returns.

Second, we notice that the SCGs are characterized by significant sparsity, with each firm typically connected to only a few direct suppliers or customers. The average degree for firms in any given industry varies between 0.2 and 1.6 across different years,[18] reflecting the concentrated nature of supply chain relationships. For example, in 2023, there are approximately 6,300 firms in the Manufacturing sector and 10,100 total edges that connect to firms in the sector, and an average degree of 1.6. This sparsity is crucial for model design, as it limits the effective range of influence any given firm's supply chain can exert on others.

The sparsity of the SCGs also has important implications for the choice of GNN layer depth d. A deeper network (higher d) enables the model to capture broader connections, but in a sparse network, this may lead to noise propagation, as information from more distant nodes may become less relevant. Thus, the sparse structure of the SCG suggests a careful balance in setting d to capture relevant dependencies without introducing noise. The sparsity observed in this dataset also highlights the challenges of maintaining connectivity in a dynamic, expanding network, as many firms maintain only a handful of key relationships within the larger supply chain network.

Lastly, the SCGs show notable cross-sectional differences across industries, particularly in the

[18] We report the detailed data in the two leading industries of Manufacturing and Wholesale & Retail Trade in Table A.1.

concentration and connectivity patterns of firms. As visible in Figure 2, most nodes in our data belong to the Manufacturing sector, followed by Wholesale & Retail Trade. This distribution reflects the supply chain's industrial concentration, where manufacturing firms often play central roles with numerous connections, while firms in other sectors may act as intermediaries or endpoints with fewer links. The average degree also varies slightly across sectors, underscoring the need to use ML to extract more complex cross-sectional differences. As can be seen in Table A.1, for instance, Manufacturing firms typically exhibit approximately equal or higher average degrees compared to firms in sectors like Wholesale & Retail Trade (firms in both industries have average degree of 1.6 in 2023), which is driven by the data disclosure rule in SEC SFAS 131 that only relationships $> 10\%$ are disclosed. Therefore, even though Wholesale & Retail Trade in reality may have much higher number of relationships (Agrawal and Osadchiy, 2024; Agca et al., 2022) our noisy and incomplete data forces us to retrieve cross-sectional differences from other means such as covariance of firm characteristics and returns conditioned on the SCG.

3.2 Asset Pricing Model

This section details our model's estimation procedure, interpretative framework, and performance metrics for firm-level return estimates and asset pricing factor performance.

3.2.1 Estimation

In this subsection, we refer to the subset of data from January 1977 to December 2006 as the training dataset, denoted by $\mathcal{D}_{\text{train}}$, and the data from January 2007 to December 2023 as the test dataset, denoted by $\mathcal{D}_{\text{test}}$. The estimation of our model parameters and asset pricing factors follows the procedure described in §2.5. The training of our model is based on minimizing the mean squared error (MSE) between estimated and actual returns, which we formally describe in (6). [19]

We intentionally shuffle the data during the training process to expose the model to a mixed sample of firms and time periods. Our shuffling strategy ensures that each mini-batch contains data from different periods and firms, allowing the model to learn generalized patterns across the full range of conditions within $\mathcal{D}_{\text{train}}$. By doing so, the model's gradient descent algorithm is less likely to overfit to specific time periods or firm-specific trends and is instead encouraged to capture the broader structural patterns in the supply chain data.

In the estimation process, we ensure that the model has reached convergence before proceeding to evaluation. Convergence is verified by monitoring the training loss as a function of the number of batches processed. Specifically, we plot the training loss on the y-axis against the number of batches on the x-axis, with each line in the plot representing a different model configuration.

[19] We conduct a comprehensive grid search to determine the optimal hyperparameters for our neural network model by minimizing MSE. Grids are constructed as a cartesian product of each parameter's individual choice set. The hyperparameters and grids we search are: epochs - [2, 3, 6], dropout rate - [0.2, 0.5, 0.8], attention heads - [1, 2, 3], output layer dimension - [40, 50, 60]. The selected parameters are as follows: 6 epochs, a dropout rate of 0.5 to control overfitting, an output layer dimension of 40, and 2 attention heads.

As shown in Figure A.1, each model configuration exhibits a declining training loss over the initial batches, eventually stabilizing as the model converges. The different colors represent various model configurations, allowing us to compare their convergence behaviors and select the configuration that best balances training efficiency and loss minimization. This plot provides a visual confirmation that the training process has stabilized, as indicated by the loss values plateauing over the final epochs. Convergence is observed when the training loss remains relatively constant across successive batches, suggesting that the model has minimized the objective function sufficiently given the data and selected hyperparameters.

3.2.2 Benchmarks

But exactly how much performance change is contributable to each part of our proposed return explanation model? Recall we use both the new data of supply chain graphs, and a new model of GNN compared to the previous literature on empiricaly asset pricing factor zoo. This can be decomposed to first asking: are the higher-order relationships in the supply chain graphs helpful for explaining returns? Secondly, conditional on using supply chain data, how much improvement can be obtained by using our novel graph neural network model over the widely used linear models? To showcase each source of improved estimation, we carefully construct benchmark comparisons by utilizing several well-established factor models in asset pricing:

- **Fama-French Five-Factor Model (FF5):** Fama and French (2015) propose a widely accepted model that includes market, size, value, profitability, and investment factors.
- **Principal Component Analysis (PCA):** A factor model that captures risk premiums through principal components, as outlined by Giglio and Xiu (2021).
- **Risk Premium Principal Component Analysis (RP-PCA):** A generalized factor model that introduces additional parameter to mix the first and second-moment objective in estimating PCA, as introduced by Lettau and Pelger (2020).
- **Neighboring Characteristics (NC)** A model that takes arithmetic average of all firm characteristics and estimates linear models. That is, for firm i, we take nodes that are in $\mathcal{N}_t(i)$ and compute the average $\bar{\mathbf{x}}_{it}$:

$$\bar{\mathbf{x}}_{it} = \frac{1}{|\mathcal{N}_t(i)|} \sum_{j \in \mathcal{N}_t(i)} \mathbf{x}_{jt}. \tag{15}$$

Then we estimate LASSO (Tibshirani, 1996) and Ridge (Hoerl and Kennard, 1970) model based on the neighboring $\bar{\mathbf{x}}_{it}$ and the firm's own characteristics $\mathbf{x}_{it}$.

$$\begin{aligned} \hat{\theta}^{\text{LASSO}} &= \text{argmin}_\theta \mathbb{E}_{(i,t)\sim\mathcal{D}_{\text{train}}} \left[\left(Y_{it} - [\mathbf{x}_{it} \quad \bar{\mathbf{x}}_{it}]^\top \theta \right)^2 \right] + \lambda_1 \|\theta\|_1, \\ \hat{\theta}^{\text{Ridge}} &= \text{argmin}_\theta \mathbb{E}_{(i,t)\sim\mathcal{D}_{\text{train}}} \left[\left(Y_{it} - [\mathbf{x}_{it} \quad \bar{\mathbf{x}}_{it}]^\top \theta \right)^2 \right] + \lambda_2 \|\theta\|_2^2, \end{aligned} \tag{16}$$

where the penalty hyperparameters λ_1 and λ_2 are selected from cross-validation.

We highlight the NC benchmark models as examples where we are using a simpler model than GNN, while conditioning on the supply chain graph information. If our machine learning approach produces better empirical performance, it would be evidence that widely used linear model are not flexible enough to capture the rich higher-order relationships that exist in supply chains to explain average returns.

These benchmark models serve as baselines for evaluating the accuracy and economic relevance of our GNN-derived factors. By comparing our model's performance against these established frameworks, we demonstrate the unique value of incorporating supply chain network information into asset pricing.

We also use the NC benchmark model to generate NC asset pricing factors $\mathbf{F}_t^{\text{NC-LASSO}}$ and $\mathbf{F}_t^{\text{NC-Ridge}}$, in order to contrast against our supply chain asset pricing factors $\mathbf{F}_t$. Therefore we need to transform NC into forms similar to (10) by running through our asset pricing factor construction procedure in Section 2.5. To align NC with GNN's embedding, we use the linear transformed concatenated firm-characteristics from either the estimated LASSO or Ridge as the embedding:

$$\tilde{\mathbf{x}}_{it}^{\text{NC-LASSO}} = \hat{\theta}^{\text{LASSO}} \odot [\mathbf{x}_{it} \quad \bar{\mathbf{x}}_{it}]^\top, \quad \tilde{\mathbf{x}}_{it}^{\text{NC-Ridge}} = \hat{\theta}^{\text{Ridge}} \odot [\mathbf{x}_{it} \quad \bar{\mathbf{x}}_{it}]^\top, \tag{17}$$

where $\odot$ is the element-wise multiplication operator[20]. With $\tilde{\mathbf{x}}_{it}^{\text{NC-LASSO}}$ and $\tilde{\mathbf{x}}_{it}^{\text{NC-LASSO}}$, we can now treat them similarly as $\tilde{\mathbf{x}}_{it}$ and follow the steps in Section 2.5 to compute asset pricing factors $\mathbf{F}_t^{\text{NC-LASSO}}$ and $\mathbf{F}_t^{\text{NC-Ridge}}$. If the portfolios formed from these NC factors generate lower Sharpe ratios than our GNN-based factors, we have further evidence that the flexible model forms of GNN are better suited for capturing the rich relationships in the supply chain data for the purpose of asset pricing.

3.2.3 Model Performance on Firm-Level Return Estimates

We now turn to show the better performance of Transformer GNN taking advantage of SCG and firm characteristics data jointly, as reported in Table 1. For firm-level monthly excess returns, we evaluate GNN models by comparing root mean squared error (RMSE) and mean absolute pricing error (MAPE) defined as:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(i,t)\sim\mathcal{D}} (Y_{it} - \hat{Y}_{it})^2}, \quad \text{MAPE} = \frac{1}{|V_{\mathcal{D}}|} \sum_{i\sim\mathcal{D}} |\frac{1}{T_i} \sum_{i\sim\mathcal{D}} (Y_{it} - \hat{Y}_{it})| \tag{18}$$

where $\mathcal{D}$ refers to either the in-sample period of $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{test}}$, $|\mathcal{D}|$ refers to the cardinality of month-firm pairs in $\mathcal{D}$, T_i refers to the number of months for firm i, and $|V_{\mathcal{D}}|$ refers the number of firms in that period. The RMSE and MAPE capture different aspects of model performance as

[20] For $\tilde{\mathbf{x}}_{it}^{\text{NC-LASSO}}$, we would have many zero entries due to LASSO's sparsity. When running the PCA in Step 2 of Section 2.5, we will drop those dimensions with zero entries induced by LASSO.

they focus on different moment conditions with distinct interpretation: the RMSE reflects variation across the panel of returns, while the MAPE captures the unexplained average returns across the cross-section of firms.

We compare our model against established benchmark models, including the Fama-French Five-Factor model (FF5), Principal Component Analysis (PCA), Risk Premium Principal Component Analysis (RP-PCA), and two Neighboring Characteristics (NC) models that consider only 1-hop neighborhood in SCG and estimated with LASSO and Ridge. We report both in-sample and out-of-sample[21] performance metrics, as shown in Table 1. The reported models use the optimal hyperparameters that are selected from grid search as described in §3.2.1. All models have reached convergence in our training, as reported in Figure A.1.

First, we find direct empirical evidence learning high-order relationships can help firm-level return estimation, as we see in Table 1 that the size of local subgraph (parameter d) impacts model performance. For example, as we increase d from $d = 1$ to $d = 6$, the in-sample MAPE improves by 17% while the out-of-sample MAPE also improves by 17%. At the same time, both in- and out-of-sample RMSE is improved as we increase from $d = 1$ to $d = 6$. Recall that the parameter d also plays a central role in the GNN architecture: d corresponds to the number of message-passing layers within the network, thus controlling the neighborhood radius over which each node aggregates information. When $d = 1$, each node aggregates information solely from its immediate neighbors, limiting the model to a localized view of the network. While this approach aligns with conventional methods that often use hand-crafted features implicitly restricted to first-order neighborhoods, it does not fully leverage the GNN's capacity to capture more complex supply chain dependencies. The results show that a $d = 1$ configuration produces higher out-of-sample RMSE and MAE compared to configurations with larger d values, suggesting that a single-layer neighborhood is inadequate for capturing the complex relational patterns within the supply chain.

Second, performance may decline if d is increased excessively. For instance, as we increase d from $d = 6$ to $d = 10$ in Table 1, the in-sample MAPE worsens by 14%, the out-of-sample MAPE worsens by 3%, while their RMSE performances remain close. This pattern highlights a fundamental bias-variance trade-off in GNN architectures, as is the case for many statistical learning problems. As d grows, the model leverages information from increasingly distant firms to produce embeddings for the firm at hand, introducing noise and potentially masking the influence of more relevant local connections. This phenomenon, known as"gradient vanishing" and "oversmoothing" (Chen et al., 2020) occurs when the relevance of close neighbors is diluted by signals from less pertinent distant nodes. As a result, higher values of d can force the GNN to unsuccessfully learn the harder-to-fit high-order patterns in the supply chain graph, capturing noise instead of meaningful structural patterns, and thus lead to reduced generalization in out-of-sample estimates. As such, we should

[21]For FF5, PCA and RP-PCA, we use in-sample period data to fit time series regressions for each firm, and hold those models as fixed for out-of-sample model evaluation. For firms that do not exist in $\mathcal{D}_{train}$ but exist in $\mathcal{D}_{test}$, we use a pooled panel regression estimated in-sample for the out-of-sample evalutions of those new firms. For GNN, the model is invariant to the firm index as long as the firm's position in the SCG is available, so we do not need special treatment firms that exist only in $\mathcal{D}_{test}$.

Table 1: Firm-level performance of GNN and benchmarks.

Model Name	In-sample		Out-of-sample	
	RMSE	MAPE	RMSE	MAPE
TransformerGNN ($d = 1$)	0.1344	0.0166	0.1606	0.0233
TransformerGNN ($d = 2$)	0.1327	0.0140	0.1593	0.0198
TransformerGNN ($d = 3$)	0.1325	0.0137	0.1591	0.0198
TransformerGNN ($d = 6$)	0.1325	0.0137	0.1590	0.0193
TransformerGNN ($d = 10$)	0.1327	0.0156	0.1589	0.0199
FF5	0.1251	0.0142	0.1689	0.0221
PCA ($k = 5$)	0.1151	0.0100	0.1605	0.0206
RP-PCA ($\gamma = 10, k = 5$)	0.1152	0.0095	0.1651	0.0207
NC (LASSO)	0.1228	0.0141	0.1570	0.0220
NC (Ridge)	0.1229	0.0142	0.1570	0.0221

This table reports the performance of our GNN models with different d, and benchmark models of FF5, PCA, RP-PCA and NC with LASSO and Ridge (defined in (16)). The NC parameters are the best LASSO and Ridge regressions. The k parameter for both PCA and RP-PCA refers to the number of latent factors. The RP-PCA parameters γ indicates the weights in a modified PCA objective.

opt for using a moderately sized model with d between 2 and 10 to balance the bias-variance trade-off.

Lastly, the GNN models such as the TransformerGNN ($d = 6$) model surpass the performance of FF5, PCA, and RP-PCA benchmarks (in the lower half of Table 1) in the out-of-sample period, while in-sample they are overfitting to the data. For instance, on the out-of-sample set, our recommended model produces 13% and 6% lower MAPE compared to FF5 and PCA models, respectively, as well as lower RMSE. When compared to NC models, it is noteworthy that our recommended model has better MAPE but worse RMSE. One explanation is that the GNN is learning from a much more expressive model class, so its second-moment error as measured by RMSE would be higher. At the same time, asset pricing is not just about the second moment; therefore, we need to further investigate the asset pricing performance in §3.2.4 when we form tradeable portfolios using our firm-level estimation models.

Our analysis on the firm-level estimation results underscores the advantages of the GNN architecture in expanding traditional asset pricing models. Based on the above analysis, we can conclude that a too simple GNN architecture ($d = 1$) is not sufficient to capture the nontrivial higher-order relational information from the supply chain network. Nevertheless, which GNN architecture ($d > 1$) achieves the optimal bias-variance tradeoff requires further analysis. In the following discussions, we shed light on the benefits of model complexity in capturing these complex relationships, and recommend a model that achieves balances in bias-variance tradeoff and has low sensitivity to noises in SCG.

3.2.4 Asset Pricing Factor Performance by Sharpe Ratio

We propose procedures that compute supply chain asset pricing factors that can form tradeable portfolios. To dissect these factors' risk-return trade-off, we examine their Sharpe ratio (SR), varying both the choice of supply chain sub-graph size (d) and the number of factors (K). For the out-of-sample (OOS) period $\mathcal{D}_{test}$, we compute our supply chain factors out-of-sample $\mathbf{F}_{\text{OOS}}$, which are excess returns, and then follow classical literature to compute out-of-sample max Sharpe ratio of:

$$\text{SR}_{\mathbf{F},\text{OOS}} = \mathbf{\Sigma}^{-1}_{\mathbf{F},\text{INS}} \mu_{\mathbf{F},\text{OOS}}, \tag{19}$$

where both in-sample (INS) period out-of-sample period are used, $\mathbf{\Sigma}_{\mathbf{F},\text{INS}}$ is the covariance matrix estimated strictly using $\mathbf{F}_{\text{INS}}$ from the in-sample period, and $\mu_{\mathbf{F},\text{OOS}}$ is the out-of-sample estimated mean of $\mathbf{F}_{\text{OOS}}$.

We report in Figure 3 the SR values for each choice of d and K. Since our factors have projected out the PCA factors, a positive SR would indicate that we have successfully captured risks unexplained by the firm-characteristics summarized by a PCA model. In other words, the out-of-sample SR reflects the unique estimation power and economic relevance of the supply chain asset pricing factors, with higher SR values indicating stronger performance.

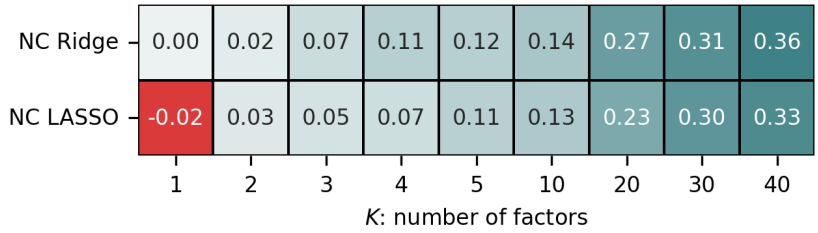
We find comprehensive and compelling evidence in Figure 3 that our supply chain-based asset pricing factors provide unique explanatory power for returns, capturing independent sources of risk beyond those explained by traditional firm characteristics. Each row in the subfigure (a) corresponds to a choice of supply chain sub-graph size (d), and each column in the subfigure (b) corresponds to a choice of number of factors K. Notably, any positive Sharpe ratio in subfigure (a) indicates that our supply chain factors identify distinct, uncorrelated risks. By construction, these factors are orthogonal due to the principal components that summarize firm characteristics, suggesting that positive Sharpe ratios reflect unique information extracted through the convolution of supply chain relationships with firm-specific attributes. This process, enabled by the GNN architecture, reveals patterns of risk and return variation that are not captured by models relying exclusively on firm characteristics.

Firstly, we find that the estimation power of the supply chain factors improves as we learn higher-order relationships with increasing d. Recall that when $d = 1$, the GNN only picks up 1-hop neighboring firms in the supply chain; when $d = 6$, the model can traverse by 6 hops and learn much more sophisticated patterns in the firms adjacent to the target firm. The Sharpe ratio performance in Figure 3 validates the virtue of complex models with higher d: relative to factors from the model with $d = 1$, we see the $d = 6$ model yields asset pricing factors that achieve 38% higher Sharpe ratio and the $d = 10$ model 54% higher Sharpe ratio, given fixed $K = 5$. This suggests that we can form portfolios with better mean-variance efficiencies when we use a GNN model that takes more complex supply chain graph relationships into account.

Secondly, we observe that most of the estimation gains are concentrated in the top 5 factors.

Figure 3: Sharpe Ratio Heatmaps (Out-of-sample).

(a) Factors by TranformerGNN, with PCA as $\mathbf{z}_t$



(b) Factors by Neighboring Characteristics, with PCA as $\mathbf{z}_t$

We report the out-of-sample Sharpe ratio of our asset pricing factors with varying number of layers (d) and increasing number of factors (K) in sub-figure (a), conditioned on PCA. We also draw comparisons against sub-figure (b) with Neighboring Characteristics (NC) factors from 1-hop neighboring firms' characteristics, constructed using (15), (16) and (17), conditioned on PCA.

For instance, when $d = 6$, the top 5 factors yield a Sharpe ratio of 0.18, while the full set of 40 factors achieves a Sharpe ratio of 0.34. This result implies that the top 5 factors account for approximately 53% of the total Sharpe ratio, illustrating a long-tailed distribution where a limited number of dominant factors explain a substantial portion of returns, while additional factors provide incremental gains. This long-tailed behavior suggests that the primary estimation power is concentrated in a select set of supply chain relationships, with the remaining factors capturing finer, less impactful variations in the data.

Finally, our supply chain factors exhibit a notable advantage over the Neighboring Characteristics (NC) baseline models as evidenced by their higher out-of-sample Sharpe ratios, which demonstrates the benefit of learning nonlinear higher-order relationships. Recall that NC admits a linear combination of characteristics of firms that are in the 1-hop neighborhood of a target firm, as estimated in (16). The improvements of our model over NC are strong. For example, when $d = 6$, the top 5 factors derived from the GNN-based supply chain model produce a Sharpe ratio of 0.18, which is 50% higher than the Sharpe ratio of 0.12 achieved by the NC Ridge model and 64% higher than the 0.11 Sharpe ratio of the NC Lasso model. This superior performance of the GNN-based approach underscores the benefit of integrating detailed relational data from the supply chain network. By leveraging the convolution of supply chain structure with firm attributes, the GNN captures nuanced, multi-layered dependencies that simpler neighborhood-averaging approaches cannot account for, demonstrating the unique value of the GNN architecture in constructing robust asset pricing factors.

In summary, we have shown that our supply chain factors contribute uniquely to asset pricing and risk management, as the out-of-sample Sharpe ratios of their portfolios exhibit superior performance. Moreover, higher-order relationships learned by higher d contribute positively to the mean-variance efficiency. The supply chain factors also have the stylized concentrated pattern in empirical finance that the leading 5 factors capture more than half of the gain in Sharpe ratio, while the remaining factors have long-tailed contributions.

4 Understanding the Supply Chain Factors and GNN

In this section, we discuss the economic interpretability of our proposed GNN model and supply chain asset pricing factors. First, we perform asset pricing regression on test assets and report t-statistics. Second, we design a Monte Carlo experiment to assess GNN estimation sensitivity to small changes in the SCG by removing edges stratified by degree centrality. In Figure 5, we report correlations between our asset pricing factors and the Fama-French factors.

4.1 Regression Analysis with Test Portfolios

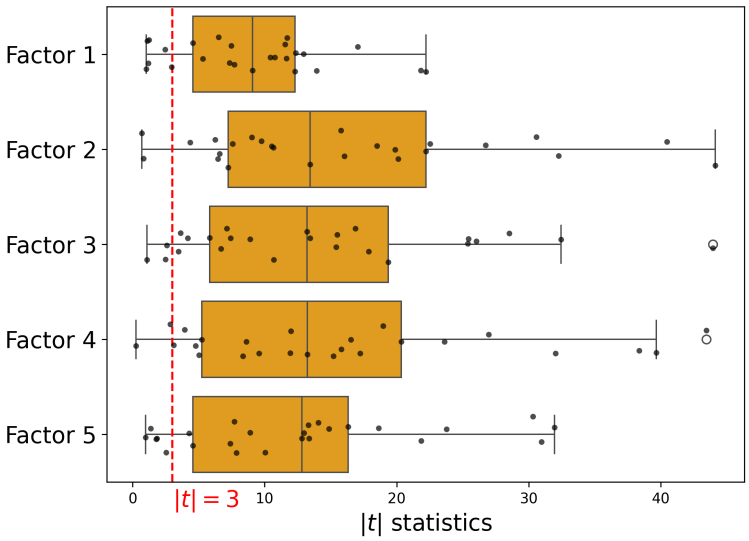
We examine the significance of our asset pricing factors by regressing them on the Fama-French 25 double-sorted Size and Book-to-Market portfolios, available from the Kenneth French's website. Specifically, for our supply chain factors $\mathbf{F}_t$, we estimate the following regression model:

$$\mathbf{r}_{pt} = \beta_p^\top \mathbf{F}_t + \eta_p^\top \mathbf{z}_t + \tilde{\varepsilon}_{pt}, \tag{20}$$

where $\mathbf{r}_{pt}$ denotes the return of portfolio p at time t, β_p captures the exposure of the portfolio to our supply chain factors $\mathbf{F}_t$, η_p captures the exposure of the portfolio to control factors $\mathbf{z}_t$, and $\tilde{\varepsilon}_{pt}$ are zero-mean idiosyncratic noises. We report the absolute values of t-statistics for each $\hat{\beta}_p$ to evaluate the statistical significance of our factors, assessing their ability to explain return variations

across portfolios. [22] We run these regressions both in-sample and out-of-sample to see whether the factors' significances persist.

Figure 4: Distribution of t-statistics across time-series regressions on test assets.



The different histograms refer to different supply chain factors. The x-axis shows the absolute values t-statistics of time-series regressions in (20). The y-axis corresponds to different supply chain asset pricing factors. The factors are constructed with TransformerGNN($d = 6$) model. The boxplot body's left, middle and right bars correspond to 25-, 50-, and 75-th percentiles.

We report the time-series regression results by taking absolute values of the t-statistics associated with the leading 5 factors in $\mathbf{F}_t$ in Figure 4 with boxplots, with FF5 as $\mathbf{z}_t$ control variables. As is evident in Figure 4, over 83% (104 out of 125) of the t-statistics are significant and above the recommended $|t| = 3$ line of Chen and Zimmermann (2022) across the test portfolios. Therefore, we find significant statistical evidence of our supply chain factors in explaining the test assets' returns.

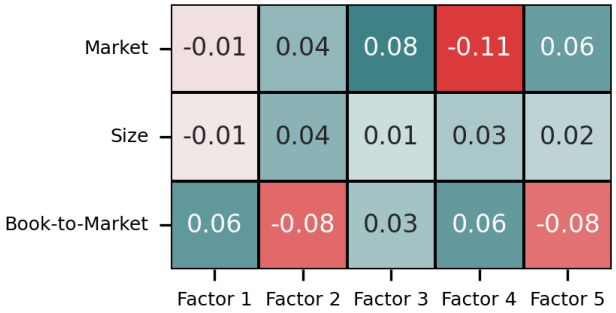
Understanding the influence of firm characteristics on our results is essential for interpreting the economic significance of our GNN-derived factors. Our goal is to identify which characteristics contribute most to the embeddings and to quantify the sensitivity of our asset pricing factors to specific characteristics. We approach this analysis using two two complementary methods. To examine how traditional firm characteristics relate to the GNN-derived embeddings $\boldsymbol{\phi}_{it}$, we compute the Pearson correlation coefficient between our factors $\mathbf{F}_t$ and well-known Fama-French three factors (FF3); see Fama and French (1993).

Correlation analysis helps to interpret each of the supply chain asset pricing factors in economic terms, revealing whether the GNN-derived features are composed of common asset pricing factors, or beyond simple combinations of them. Since our supply chain factors are residualized portfolio returns of principal directions of the GNN's embeddings covariance, the correlation serve as a more interpretable stopgap to understanding how our supply chain factors are unique. We report

[22] Using t-statistics is a common approach to ensure reproducibility when repeatedly conducting asset pricing, which falls under the topic of multiple testing for asset pricing as discussed in Barillas and Shanken (2018); Chen and Zimmermann (2022); Pelger and Zou (2024).

our findings in Figure 5. The first observation is that none of our factors are strongly correlated with Fama-French factors, as the highest correlation in magnitude is 0.11. At the factor-by-factor level, we find that the first supply chain factor is weakly positively correlated with the Book-to-Market factor, while having small negative correlation with Market and Size. The second factor has moderate correlation with all FF3 factors, with different signs on Book-to-Market vs Market and Size. The third factor has across the board positive correlation with FF3 factors. The fourth factor has the strongest correlation with Market in the negative sign, and weak positive correlations with Size and Book-to-Market. The fifth factor again shows positive correlations with both Market and Size, and different signed correlation with Book-to-Market.

Figure 5: Correlation with Fama-French 3 factors.



The y-axis shows the three Fama-French factors (Fama and French, 1993) of Market, Size and Value. The x-axis shows the leading 5 supply chain asset pricing factors. Factors are estimated with $d = 6$ TransformerGNN with PCA as control $\mathbf{z}_t$.

4.2 Asset Pricing Sensitivity to the Supply Chain Graph

To better understand how sensitive our asset pricing factors are to small perturbations in the supply chain network, we conduct a "derivative-like" analysis by examining the effects of infinitesimal changes in the graph structure. This approach enables us to assess how local structural changes within the network influence the factors and, consequently, the estimated returns.

Our method proceeds as follows. We first partition the nodes (firms) in the graph into four different groups based on their degrees (i.e., the number of direct connections each node has), namely: $\{1\}, \{2, 3\}, \{4, 5\}, \{\geq 6\}$. These groups are defined so that we can directly interpret their economic meaning of the firm's centrality in the SCG. However, the firms are not evenly distributed among them. There are a greater number of low-degree nodes, and the groups' firm counts vary over time as firms enter and leave the market or adjust their supply chain relationships. [23] To avoid biasing the sensitivity analysis results by simply perturbing more nodes in the low-degree group with the larger number of nodes, we perturb a fixed number of 100 nodes per group in each Monte Carlo simulation, ensuring that the sensitivity measure remains comparable across degree groups.

[23] More details on these dynamics are reported in Figure A.2, where we plot a time series of the number of firms within each degree group, providing context for how degree distributions evolve.

Table 2: Mean and standard deviation of Monte Carlo experiments on SCG.

Model	Degree: 1	Degree: 2-3	Degree: 4-5	Degree: ≥6
TransformerGNN ($d = 1$)	0.91 (0.67)	1.56 (1.07)	1.70 (1.23)	2.30 (1.81)
TransformerGNN ($d = 2$)	0.88 (0.48)	1.24 (0.72)	1.31 (0.83)	1.81 (1.35)
TransformerGNN ($d = 3$)	0.66 (0.37)	0.71 (0.36)	0.77 (0.47)	1.09 (0.82)
TransformerGNN ($d = 6$)	0.29 (0.15)	0.36 (0.19)	0.41 (0.27)	0.63 (0.51)
TransformerGNN ($d = 10$)	0.61 (0.36)	0.66 (0.30)	0.75 (0.48)	1.10 (0.85)

The reported values are average sensitivity and in parenthesis the standard deviation of the sensitivity, in units of 10^{-5}. Sensitivity is defined in (21). The results are taken over $B = 100$ simulations. The columns show different groups of nodes with increasing degree centrality from left to right. The left-most column corresponds to firms with only 1 degree in SCG, whereas the right-most column corresponds to firms with ≥ 6 degrees in SCG.

We perform a Monte Carlo simulation where, in each iteration, we uniformly and randomly delete one edge connected to a randomly chosen node within each group. This process allows us to measure the change in the asset pricing resulting from the removal of a single, incidental connection in the network.

Supply chain graph sensitivity measure $\tilde{S}$ for firm i at time t is computed as the absolute difference between the original estimate and the perturbed estimate after an edge deletion:

$$\tilde{S}_{it}^{(b)} = \left| \hat{g}_i(\mathbf{X}_t, SCG_t^{(b)}) - \hat{g}_i(\mathbf{X}_t, SCG_t) \right|, \tag{21}$$

where $SCG_t^{(b)}$ is the bth perturbation of the SCG_t graph and $b = 1, 2, ..., B$ as we run our Monte Carlo experiment. The $\tilde{S}_{it}$ metric provides a straightforward way to quantify how much each estimate changes in response to edge deletions. By observing this metric over multiple Monte Carlo simulations, we obtain an intuitive estimate of the sensitivity for each degree group and model configuration.

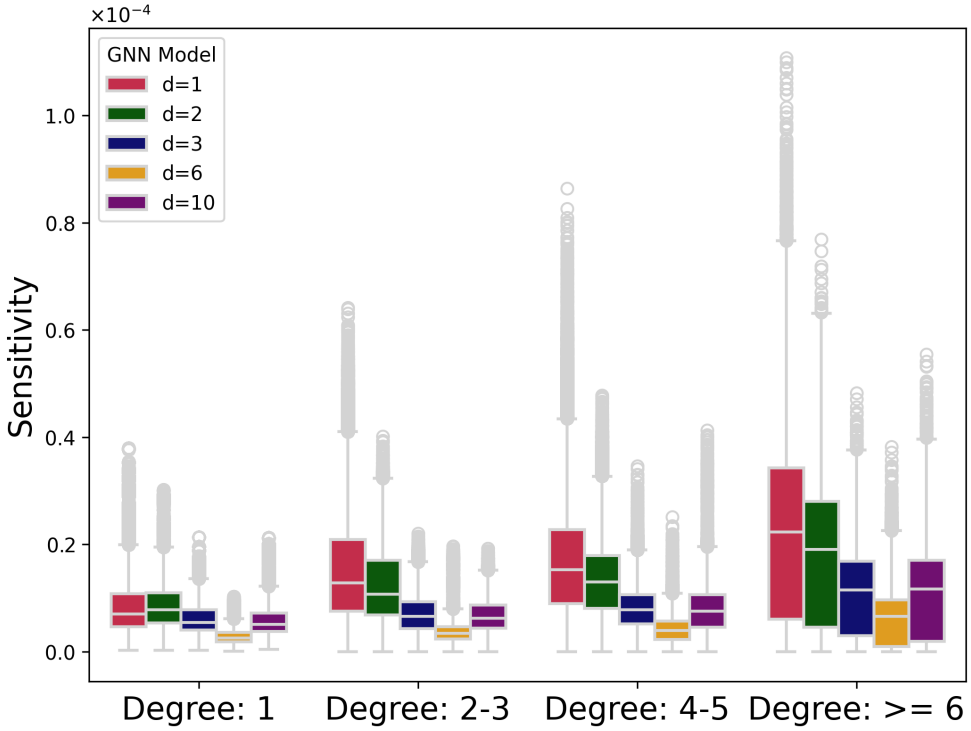
The Monte Carlo simulation is repeated B times for each group, providing a distributional estimate of the effect of edge deletions. We introduce the randomness in order to marginalize over variations due to specific edge or node choices within each group, and to compute the overall sensitivity of the factors to small changes in connectivity for firms of varying degrees.

Empirically, we report our results in Figure 6 as boxplots that show not only the median impact but also the variability of sensitivity within each degree group, providing insight into how structural changes at different levels of the supply chain network affect model robustness. We also report the mean and standard deviation of the simulation results in Table 2.

We derive two key economic insights about the stability of model estimates in response to SCG perturbations: one concerning the role of a firm's degree centrality, and the other addressing the effect of model complexity (d) on GNN.

First, regarding firm centrality: as the degree of perturbed nodes increases, the model's estimates demonstrate greater variability in $\hat{y}$, reflected both in higher average sensitivity $\tilde{S}$ and increased variance. Visually, we see the boxes of Figure 6 move higher from left to right as the

Figure 6: Sensitivity analysis of Monte Carlo experiments on SCG.



This figure shows the distributional results of sensitivity in model performance across degree groups and models when we randomly remove edges from SCG's specific group of firms. Sensitivity is defined in (21). The x-axis shows four degree groups, while the y-axis displays the distribution of $\tilde{S}_{it}$ as a result of random edge deletions. Each box plot represents the distributional variation of the edge deletion Monte Carlo simulations. The results are taken over $B = 100$ simulations and use all the data from 1977 to 2023.

degree of the firm being perturbed increases. More specifically from Table 2, with our recommended TransformerGNN of $d = 6$ model, we see the average and standard deviation of sensitivity of firms with ≥ 6 degrees is still 41% and 240% higher than that of firms with 1 degree. In models such as TransformerGNN with $d = 2$, we observe that the average sensitivity and standard deviation for firms with ≥ 6 degrees is 152% and 181% higher than that of firms with a degree of 1.

This empirical evidence aligns with the underlying economic mechanism in supply chain: we expect high-degree nodes, which represent central firms with more supply chain connections, play a critical role in the network structure. When edges of these nodes are perturbed, even small changes can ripple through the network, significantly impacting model estimates. By contrast, low-degree nodes are less central and have fewer connections, leading to minimal changes in estimates with smaller variations. This suggests that the GNN embeddings of low-degree nodes are more stable, as these peripheral nodes contribute less to the overall supply chain structure and thus have limited influence on the relational dynamics captured by the model.

Secondly, regarding the role of d, our sensitivity analysis reveals that the recommended model, with carefully selected layers, exhibits the lowest sensitivity to edge perturbations. From Table 2, we see that our recommended TransformerGNN with $d = 6$ model has 65% lower average sensitivity

and 62% lower standard deviation in sensitivity than the $d = 2$ model, for the most central firms (degree ≥ 6) in the supply chain.

From an economic perspective, as d increases from small to moderate, the receptive field of each firm's supply chain model expands, allowing us to capture more complex, multi-hop dependencies within the supply chain network. However, when d increases to an even larger number such as $d = 10$, the GNN becomes excessively deep, leading to instability in the model estimation and resulting in higher variance in the estimates.

This result aligns with theoretical expectations: for each layer in the neural network, a random dropout function already introduces inherent robustness to make sure the model focuses on learning core patterns, and thus the case of GNN with $d = 6$ is indeed capturing fundamental economic nature in SCG despite noisy observations of the relationship graph. However, as we further increase the depth of GNN, we expect the classical bias-variance trade-off in statistical learning problems. This claim is supported by the evidence in Table 2 – as we increase $d = 10$, we extract more complex patterns from SCG but at the cost of larger variance: compared to $d = 6$'s model, we have 178% larger variance for the most central firms, and 476% larger variance for the firms with only one connection.

In summary, our sensitivity analysis highlights the interplay between node centrality, model complexity, and estimate stability in GNN-based models. High-degree nodes play a crucial role in maintaining the structural integrity of the supply chain network, making the model's estimates more sensitive to changes involving these nodes. In contrast, low-degree nodes demonstrate greater stability, indicating that peripheral firms have a limited impact on the dynamics captured by the GNN.

Additionally, increasing d enhances the model's ability to learn complex relationships. However, excessively large values of d result in high variance, reflecting the bias-variance trade-off discussed in Table 1.

5 Conclusion

We have developed a novel empirical framework that highlights the value of supply chain information in asset pricing. Our proposed machine learning framework, which integrates supply chain and firm characteristics nonlinearly, produces new asset pricing models with superior out-of-sample performance compared to the FF5 model and PCA-based approaches. The supply chain asset pricing factors derived from regression on test assets are statistically significant and deliver positive Sharpe ratios out-of-sample.

We highlight that the size of the supply chain local neighborhood (d) is a critical and interpretable parameter that requires careful selection. Our empirical analysis demonstrates that $d = 6$ achieves an effective balance between the bias introduced by omitting higher-order supply chain structures and the variance of the estimator. This moderate level of model complexity also exhibits reduced sensitivity to noise in the graph compared to more complex configurations.

Our findings reveal that the model's estimates are more sensitive to perturbations in the supply chain graph for firms with higher degree centrality, aligning with existing literature that associates supply chain centrality with variations in firm returns.

To extract meaningful signals from supply chain data amidst numerous firm characteristics, we employ flexible graph models that dynamically aggregate information from local subgraphs. Our proposed GNN methodology is specifically designed for this purpose.

While our primary focus has been on asset pricing, our framework extends to a wide range of economic problems involving large graphs and panel data. This includes applications such as causal inference with interference graphs and rich covariates, provided the objective can be expressed as a weighted sum of moment conditions. The parameter d serves as a data-driven mechanism for aggregating covariates from units within d-hops of a target node in a given graph.

References

Acemoglu, D., Carvalho, V. M., Ozdaglar, A., and Tahbaz-Salehi, A. (2012). The network origins of aggregate fluctuations. *Econometrica*, 80(5):1977–2016.

Agca, S., Babich, V., Birge, J. R., and Wu, J. (2022). Credit shock propagation along supply chains: Evidence from the cds market. *Management Science*, 68(9):6506–6538.

Agrawal, D. and Osadchiy, N. (2024). Inventory productivity and stock returns in manufacturing networks. *Manufacturing & Service Operations Management*, 26(2):573–593.

Ba, J. L., Kiros, J. R. K., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Babich, V. and Birge, J. R. (2021). The interface of finance, operations, and risk management. *Foundations and Trends® in Technology, Information and Operations Management*, 15(1–2):1–203.

Barillas, F. and Shanken, J. (2018). Comparing asset pricing models. *The Journal of Finance*, 73(2):715–754.

Barrot, J.-N. and Sauvagnat, J. (2016). Input specificity and the propagation of idiosyncratic shocks in production networks. *The Quarterly Journal of Economics*, 131(3):1543–1592.

Birge, J., Capponi, A., and Chen, P. (2023). Disruption and rerouting in supply chain networks. *Operations Research*, 71(2):750–767.

Brody, S., Alon, U., and Yahav, E. (2021). How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Bryzgalova, S., DeMiguel, V., Li, S., and Pelger, M. (2023). Asset-pricing factors with economic targets. *Available at SSRN*, 4344837.

Capponi, A., Du, C., and Stiglitz, J. (2024). Are supply networks efficiently resilient. *National Bureau of Economic Research, Working Paper*, (32221).

Carvalho, V. M., Nirei, M., Saito, Y. U., and Tahbaz-Salehi, A. (2021). Supply chain disruptions: Evidence from the great east japan earthquake. *The Quarterly Journal of Economics*, 136(2):1255–1321.

Cen, L., Maydew, E. L., Zhang, L., and Zuo, L. (2017). Customer–supplier relationships and corporate tax avoidance. *Journal of Financial Economics*, 123(2):377–394.

Chen, A. Y. and Zimmermann, T. (2022). Publication bias in asset pricing research. *arXiv preprint arXiv:2209.13623*.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445.

Chen, L., Pelger, M., and Zhu, J. (2024). Deep learning in asset pricing. *Management Science*, 70(2):714–750.

Clevert, D.-A. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cochrane, J. H. (2009). *Asset pricing: Revised edition*. Princeton university press.

Cochrane, J. H. (2011). Presidential address: Discount rates. *The Journal of finance*, 66(4):1047–1108.

Cohen, L. and Frazzini, A. (2008). Economic links and predictable returns. *The Journal of Finance*, 63(4):1977–2011.

Connor, G. and Korajczyk, R. A. (1988). Risk and return in an equilibrium apt: Application of a new test methodology. *Journal of financial economics*, 21(2):255–289.

Ding, W., Levine, R., Lin, C., and Xie, W. (2021). Corporate immunity to the covid-19 pandemic. *Journal of financial economics*, 141(2):802–830.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, 47(2):427–465.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56.

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Farmer, L. E., Schmidt, L., and Timmermann, A. (2023). Pockets of predictability. *The Journal of Finance*, 78(3):1279–1341.

Feng, G., Giglio, S., and Xiu, D. (2020a). Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370.

Feng, G., Giglio, S., and Xiu, D. (2020b). Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370.

Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, 33(5):2326–2377.

Giglio, S. and Xiu, D. (2021). Asset pricing with omitted factors. *Journal of Political Economy*, 129(7):1947–1990.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hamilton, W. L. (2020). *Graph representation learning*. Morgan & Claypool Publishers.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2022). Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Herskovic, B. (2018). Networks in production: Asset pricing implications. *The Journal of Finance*, 73(4):1785–1818.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hou, K., Xue, C., and Zhang, L. (2020). Replicating anomalies. *The Review of financial studies*, 33(5):2019–2133.

Hudgens, M. G. and Halloran, M. E. (2008). Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842.

International Energy Agency (2024). Analysing the impacts of russia's invasion of ukraine on energy markets and energy security. https://www.iea.org/topics/russias-war-on-ukraine.

Kelly, B., Malamud, S., and Zhou, K. (2024). The virtue of complexity in return prediction. *The Journal of Finance*, 79(1):459–503.

Kelly, B. T., Malamud, S., and Zhou, K. (2022). The virtue of complexity everywhere. *Available at SSRN 4166368*.

Kelly, B. T., Pruitt, S., and Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kozak, S., Nagel, S., and Santosh, S. (2020). Shrinking the cross-section. *Journal of Financial Economics*, 135(2):271–292.

Lettau, M. and Pelger, M. (2020). Factors that fit the time series and cross-section of stock returns. *The Review of Financial Studies*, 33(5):2274–2325.

Mohan, B., Buell, R. W., and John, L. K. (2020). Lifting the veil: The benefits of cost transparency. *Marketing Science*, 39(6):1105–1121.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

Pelger, M. and Zou, J. (2024). Selective multiple testing: Inference for large panels with many covariates. *Available at SSRN 4315891*.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. (2021). Masked label prediction: Unified message passing model for semi-supervised classification. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Sodhi, M. S. and Tang, C. S. (2019). Research opportunities in supply chain transparency. *Production and Operations Management*, 28(12):2946–2959.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Wall Street Journal (2024). New disruptions, geopolitics hang over 2024 supply chains. https://www.wsj.com/articles/new-disruptions-geopolitics-hang-over-2024-supply-chains-6fca8f1e.

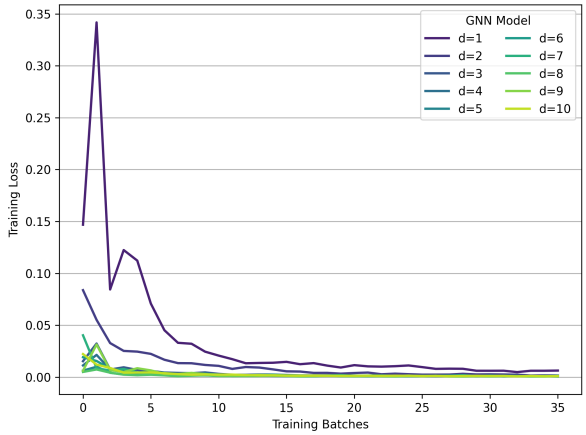
Wu, D. (2024). Text-based measure of supply chain risk exposure. *Management Science*, 70(7):4781–4801.

Appendix

A Appendix: Empirical Asset Pricing

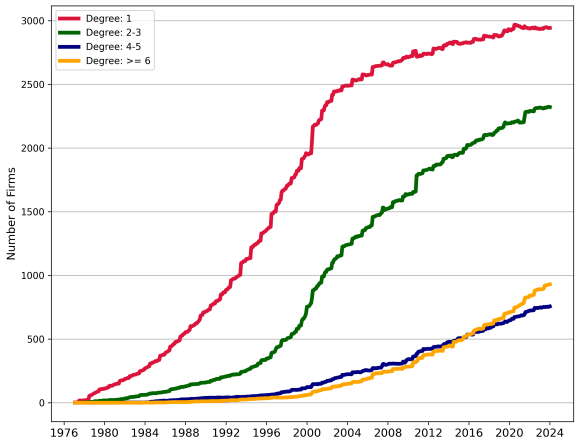
A.1 Table and Figures

Figure A.1: Training loss over batches for different model configurations.



The x-axis represents the number of batches, while the y-axis shows the training loss. Each line corresponds to a different model variant. From this figure, we can observe that trainings have converged for all the models.

Figure A.2: Number of firms in each group of degrees across time.



The x-axis represents the time, while the y-axis shows the number of firms. Each line corresponds to a different group. This figure reveals the complex temporal dynamics in the number of firms within each group. To address this, we sample an equal number of nodes from each group for our Monte Carlo experiments, ensuring that our analysis is not biased toward the group with the largest node count.

Table A.1: Summary statistics for firms in top two leading industries

Year	Manufacturing			Wholesale & Retail Trade		
	Num. links	Num. firms	Average degree	Num. links	Num. firms	Average degree
1977	35	148	0.24	5	19	0.26
1978	80	182	0.44	13	28	0.46
1979	104	204	0.51	15	30	0.50
1980	137	237	0.58	19	35	0.54
1981	179	291	0.62	21	42	0.50
1982	231	358	0.65	25	47	0.53
1983	298	415	0.72	33	58	0.57
1984	363	469	0.77	40	66	0.61
1985	448	538	0.83	56	86	0.65
1986	525	610	0.86	73	102	0.72
1987	616	686	0.90	97	129	0.75
1988	694	741	0.94	113	146	0.77
1989	789	819	0.96	143	175	0.82
1990	877	887	0.99	176	205	0.86
1991	996	994	1.00	208	239	0.87
1992	1115	1103	1.01	272	305	0.89
1993	1247	1217	1.02	316	353	0.90
1994	1377	1324	1.04	377	415	0.91
1995	1599	1495	1.07	439	476	0.92
1996	1806	1657	1.09	502	543	0.92
1997	1977	1792	1.10	569	610	0.93
1998	2147	1903	1.13	654	693	0.94
1999	2481	2117	1.17	788	784	1.01
2000	3085	2512	1.23	972	876	1.11
2001	3366	2689	1.25	1070	936	1.14
2002	3650	2841	1.28	1200	1020	1.18
2003	3929	2976	1.32	1302	1072	1.21
2004	4107	3066	1.34	1391	1127	1.23
2005	4420	3206	1.38	1559	1223	1.27
2006	4635	3313	1.40	1645	1267	1.30
2007	4830	3398	1.42	1799	1344	1.34
2008	4950	3442	1.44	1893	1385	1.37
2009	5216	3547	1.47	1993	1438	1.39
2010	5712	3728	1.53	2105	1484	1.42
2011	6162	3889	1.58	2204	1536	1.43
2012	6432	4008	1.60	2305	1583	1.46
2013	6756	4142	1.63	2435	1646	1.48
2014	6950	4223	1.65	2637	1730	1.52
2015	7398	4446	1.66	2722	1772	1.54
2016	7844	4683	1.67	2843	1826	1.56
2017	8136	4821	1.69	2938	1868	1.57
2018	8498	5007	1.70	3019	1898	1.59
2019	8940	5268	1.70	3102	1941	1.60
2020	9299	5528	1.68	3239	2007	1.61
2021	9607	5805	1.65	3345	2066	1.62
2022	10077	6055	1.66	3405	2094	1.63
2023	10147	6298	1.61	3414	2130	1.60

We report the number of edges of all nodes and the number of nodes in the corresponding industry, as well as average degree. The number of nodes reflect the number of firms and the number of edges reflect the number of observed supply chain links in our data.

B Appendix: Neural Network Layers

B.1 Transformer Architecture

In our model, we enhance the GNN learning by incorporating a Transformer architecture proposed in Shi et al. (2021), which allows the GNN to more effectively capture complex relationships within the supply chain network. A multi-head Transformer-based approach introduces cross-attention mechanisms that assign different weights to each node's neighboring messages, dynamically learning the importance of various connections.

For each node i at layer l with $l = 1, 2, ..., d$, the H-multi-head Transformer mechanism computes a set of attention weights that determine the importance of each neighbor $j \in \mathcal{N}(i)$ in the update process. The Transformer-based computation for node i can be represented as follows:

- **Message with Multi-Head Attention** (f_{msg}): For a node i, the message from a neighboring node j is computed using multiple attention heads. Each head h independently computes attention weights and corresponding messages:

$$\mathbf{m}_{ij}^{(l,h)} = \text{softmax}\left(\frac{(\mathbf{W}_Q^{(l,h)}\mathbf{h}_i^{(l)})^\top(\mathbf{W}_K^{(l,h)}\mathbf{h}_j^{(l)})}{\sqrt{d_k}}\right)\mathbf{W}_V^{(l,h)}\mathbf{h}_j^{(l)}, \tag{22}$$

  where $\mathbf{W}_Q^{(l,h)}, \mathbf{W}_K^{(l,h)}, \mathbf{W}_V^{(l,h)}$ are the learnable weight matrices for the query, key, and value transformations for head h in layer l, $\mathbf{h}_i^{(l)}$ and $\mathbf{h}_j^{(l)}$ are the feature vectors of nodes i and j, and d_k is the dimension of the key vectors. The softmax function normalizes the attention scores across all neighbors.

- **Aggregation Across Attention Heads** (AGG): The aggregated message for node i is computed by concatenating the outputs of all attention heads and applying a learnable linear transformation:

$$\mathbf{m}_i^{(l)} = \mathbf{W}_O^{(l)} \cdot \text{concat}\left(\mathbf{m}_i^{(l,1)}, \mathbf{m}_i^{(l,2)}, \ldots, \mathbf{m}_i^{(l,H)}\right), \tag{23}$$

  where H is the number of attention heads, $\mathbf{m}_i^{(l,h)} = \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}^{(l,h)}$ is the aggregated message for head h, and $\mathbf{W}_O^{(l)}$ is a learnable matrix that combines the outputs of all heads for lth layer.

- **Update** (f_{upd}): The node state is updated using a feedforward neural network (FFN) with residual connections and normalization, as is typical in transformers:

$$\mathbf{h}_i^{(l+1)} = \text{LayerNorm}\left(\mathbf{h}_i^{(l)} + \text{FFN}\left(\text{LayerNorm}(\mathbf{h}_i^{(l)} + \mathbf{m}_i^{(l)})\right)\right), \tag{24}$$

  where the first residual connection combines the previous node state $\mathbf{h}_i^{(l)}$ with the aggregated message $\mathbf{m}_i^{(l)}$, and the second residual connection refines the updated state through a position-wise feedforward network. He et al. (2016) proposed the residual connection (adding $h_i^{(l)}$) to

help mitigate gradient vanishing issues, facilitating the learning of deeper networks, and it is well-recognized as a popular technique in ML.

Our choice of FFN leverages ELU activation function and the LayerNorm regularization (Ba et al., 2016) to compute the aggregated information at each node by applying a series of transformations. The FFN for node i at layer l can be written as:

$$\mathrm{FFN}(\mathbf{z}_i^{(l)}) = \mathbf{W}_2^{(l)} \cdot \mathrm{ELU}(\mathbf{W}_1^{(l)} \mathbf{z}_i^{(l)} + \mathbf{b}_1^{(l)}) + \mathbf{b}_2^{(l)},$$

where:

- $\mathbf{z}_i^{(l)}$: Input to the FFN for node i, typically $\mathbf{z}_i^{(l)} = \mathrm{LayerNorm}(\mathbf{h}_i^{(l)} + \mathbf{m}_i^{(l)})$, combining the previous node state $\mathbf{h}_i^{(l)}$ and the aggregated message $\mathbf{m}_i^{(l)}$.
- $\mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}$: Learnable weight matrices in the FFN layers; $\mathbf{b}_1^{(l)}, \mathbf{b}_2^{(l)}$: Learnable bias vectors.
- $\mathrm{ELU}(x)$: "Exponential linear unit" is a nonlinear entry-wise activation function defined in Clevert (2015) of

$$\mathrm{ELU}(x) = \begin{cases} x & x > 0 \\ \exp(x) - 1 & x \leq 0 \end{cases}. \tag{25}$$