A differentiable model of supply-chain shocks

Saad Hamid[1]*
saad.hamid@aioilab-oxford.eu

José Moran[2,3]
jose.moran@macrocosm.group

Luca Mungo[2,3]
luca.mungo@macrocosm.group

Arnau Quera-Bofarull[2]
arnau@macrocosm.group

Sebastian Towers[2,4,5]
sebastian@macrocosm.group

[1] Aioi R&D Lab [2] Macrocosm
[3] Institute for New Economic Thinking at the Oxford Martin School, University of Oxford
[4] FLAIR, Foerster Lab for AI Research, University of Oxford
[5] Department of Engineering, University of Oxford

Abstract

Modelling how shocks propagate in supply chains is an increasingly important challenge in economics. Its relevance has been highlighted in recent years by events such as Covid-19 and the Russian invasion of Ukraine. Agent-based models (ABMs) are a promising approach for this problem. However, calibrating them is hard. We show empirically that it is possible to achieve speed ups of over 3 orders of magnitude when calibrating ABMs of supply networks by running them on GPUs and using automatic differentiation, compared to non-differentiable baselines. This opens the door to scaling ABMs to model the whole global supply network.

1 Introduction

Supply-chain disruptions cause macroeconomic damages [7], and yet, despite an established research tradition [17, 1, 3], they remain difficult to model realistically [9]. This challenge arises from fundamental limitations in existing approaches. Traditional economic models rely on *comparative statics* and cannot capture the explicitly dynamical phenomena that characterize disruption propagation through supply networks. Many modern machine-learning methods are similarly constrained by data scarcity: the historical record contains only a few disruptions (e.g., pandemics or natural disasters) on which to train, making it difficult to generalize across different types of shocks or network structures.

Agent-based models (ABMs) [2] offer a promising alternative to these limitations. ABMs are a computational paradigm where a collection of agents and their interactions are modelled explicitly in a simulator. Unlike statistical methods, ABMs provide mechanistic models that recover macroscopic quantities from the bottom-up through emergent processes. ABMs are therefore a natural choice for modelling supply chains disruptions, by capturing the micro-dynamics of each firm's production and inventory management [12, 13, 14].

However, ABMs are hard to calibrate. These models contain latent parameters that must be calibrated to match real-world observations according to some preferred metric. For example, firm productivity

*Authors listed alphabetically.

1st Workshop on Differentiable Systems and Scientific Machine Learning @ EurIPS 2025

may not be directly observable and must be inferred from macroscopic quantities. Calibration requires evaluating the simulator at multiple points in parameter space — potentially thousands or millions of times — incurring very high computational costs when the simulator is slow to run. Furthermore, the intractable likelihood function of ABMs prevents straightforward application of Bayesian calibration methods, hindering proper uncertainty quantification.

Several approaches have emerged to mitigate these calibration challenges. These range from developing surrogate methods that mimic ABM outputs at greater speed [10, 25], to neural-based approaches that directly learn likelihood or posterior distributions [11]. There has also been significant effort in tensorizing ABMs to leverage GPU acceleration, achieving substantial speedups [8]. Furthermore, applying automatic differentiation to ABMs enables fast sensitivity analyses that can greatly accelerate calibration methods [8, 24].

In this work, we present a supply chain ABM implemented in `JAX` [6], a differentiable programming, high-performance numerical computing framework for Python. We demonstrate how supply chain models can be made amenable to tensorization across thousands of firm agents, achieving substantial speed-ups compared to traditional implementations. Furthermore, by leveraging `JAX`'s automatic differentiation (AD) capabilities and its probabilistic programming library `Numpyro` [20], we apply generalized variational inference to perform efficient ABM calibration with proper uncertainty quantification over thousands of parameters.

2 Background and Related Work

Attempts to model production networks date back to Leontief's Input-Output framework [17], later extended to dynamic and firm-level contexts by Long and Plosser [18], Acemoğlu *et al.* [1], and Baqaee and Farhi [3, 4]. These models, widely used in the mainstream economic literature, study shock propagation through *comparative statics*: they compute equilibrium allocations before and after an exogenous productivity shocks, and compare outcomes such as aggregate output. While useful to understand the effects of network topology on shock propagation, this approach misses dynamical features like inventory adjustment or time-dependent recovery.

To overcome this, researchers have turned to agent-based models (ABMs) which model supply-chain dynamics from the bottom-up. Building on the ARIO model of Hallegatte [12, 13], recent work has replicated real-world shock propagation such as the 2011 Tohoku earthquake [14], the effect of the Covid-19 lockdowns in the UK's economy [22] or the effect on the Austrian economy of an embargo on Russian gas [21]. However, none of these works attempted a serious calibration of their models beyond parameter exploration.

Parameter estimation in ABMs has traditionally relied on Approximate Bayesian computation (ABC) methods, which sidestep intractable likelihood computation by matching summary statistics between simulated and observed data. While this approach provides a general framework for ABM calibration, it suffers from sensitivity to summary statistic selection and poor scaling in high-dimensional parameter spaces [23]. These limitations have motivated the development of more sophisticated methodological approaches.

One prominent strategy develops surrogate models that approximate ABM behavior while maintaining computational tractability and differentiability. By replacing the original ABM with these more amenable representations, researchers can apply established parameter estimation techniques, including Markov Chain Monte Carlo and gradient-based variational inference [25, 10, 19, 16].

Neural methods represent another paradigm, directly learning the mapping from simulator parameters to posterior distributions [11, 5]. These approaches offer compelling advantages through amortized inference—once trained, they provide immediate posterior estimates for new observations without retraining. However, their inability to exploit gradient information limits efficiency in high-dimensional parameter spaces.

Most recently, advances in differentiable programming have enabled a third approach that exploits the inherent computational structure of ABMs. Network-based ABMs in epidemiology have demonstrated high degrees of tensorization, achieving substantial GPU acceleration [8]. By applying relaxation techniques to discrete randomness and control flow structures, these methods integrate ABMs directly into AD frameworks, enabling end-to-end optimization without the approximation errors of surrogate models or the gradient limitations of neural approaches [24].

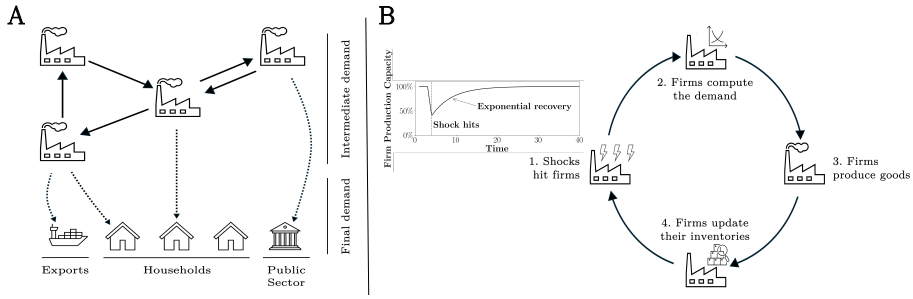

Figure 1: **A**: We model a directed production network where firms produce to serve demand from other firms, households, government, and foreign customers. **B**: In each period, an exogenous shock may limit each firm's capacity to produce, then firms receive and place orders, produce subject to input inventories and a capacity limit, and update inventories.

3 Methods

3.1 Model

We model a directed production network of M firms, where each firm i produces one good to serve demand from other firms (intermediate demand) as well as households, government, and foreign customers (final demand, d_i). Time is discrete: in each period, firms receive and place orders, produce subject to input inventories and a capacity limit, and update inventories; at the beginning of each period, an exogenous shock can decrease the maximum output that they can produce. Input requirements are encoded by a technical-coefficient matrix $\mathbf{A}$ (entries A_{ij} are units of input j required to produce one unit of output i), and inputs and outputs are tied by the formula $f\left(S_{ij}\right) = z_i \min \frac{S_{ij}}{A_{ij}}$, where z_i is the productivity and set to $z_i = 1$ in absence of a shock.

Firms try to maintain a target inventory $S_{ij}^{\text{target}} = n_i S_{ij}(0)$, enough to produce for n_i iterations if they don't receive supplies. Let total demand for good i be $D_i(t) = d_i + \sum_k O_{ki}(t)$. Firms place orders to cover final demand and close inventory gaps, with an adjustment speed $1/\tau_i$, so $O_{ji}(t) = A_{ji} D_i(t) + \left[S_{ij}^{\text{target}} - S_{ji}(t-1)\right]/\tau_i$.

The actual production is the smallest among the demand received, and the firms' production capacity, given productivity and input constraints: $x_i(t) = \min\left\{D_i(t-1),\ z_i(t) f\left(S_{ji}(t)\right)\right\}$,where the productivity follows a shock-and-recovery process $z_i(t) = \left(1 - \delta_i \exp\left(-\lambda_i (t - t^*)^+\right)\right)$ for shocked firms, with $(t - t^*)^+ = \max\{0, t - t^*\}$ and t^* the time of the shock. Inventories evolve by inflow minus use, $S_{ji}(t) = S_{ji}(t-1) + Z_{ji}(t) - A_{ji} x_i(t)$, where $Z_{ji}(t)$ denotes the realized deliveries from j to i. We initialize at a steady state (orders match D_i and inventories sit at target) and study how shocks propagate through orders, inventories, and capacity constraints.

The model produces individual time series for each firm $x_i(t)$, but in practice, we aggregate them to obtain macroeconomic aggregates that would match regional production indices. This is obtained, by defining $y_r(t) = \sum_{i \in r} x_i(t)$ where $1 \leq r \leq N$ denotes the set of firms in a given sector and region.

3.2 Calibration

Let $\mathbf{y}$ denote the aggregate observed time series of firms' outputs, $\mathbf{n} = [n_1, \ldots, n_N]$ the model parameters — the array of firms' inventory levels — with prior $p(\mathbf{n})$, and $\ell(\mathbf{y}; \mathbf{n})$ a loss function measuring the discrepancy between the observed data and the simulated results. We estimate the (generalized) posterior over parameters with Generalized Variational Inference (GVI) [15], which approximates $p(\mathbf{n} \mid \mathbf{y})$ by a distribution $q^*(\mathbf{n}) \in \mathcal{Q}$ solving

$$q^*(\mathbf{n}) \;=\; \arg\min_{q(\mathbf{n}) \in \mathcal{Q}} \; \mathbb{E}_{\mathbf{n} \sim q}[\ell(y; \mathbf{n})] \;+\; D(q(\mathbf{n}) \,\|\, p(\mathbf{n}))\,, \tag{1}$$

where $D\left(\cdot \| p\left(\mathbf{n}\right)\right)$ penalizes deviations from the prior. In practice, we parameterize $q\left(\mathbf{n}\right)$ using a set of variational parameters ϕ (e.g., $\mathcal{Q}$ can be the Gaussian family $q_\phi(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \mu, \Sigma)$ with $\phi = (\mu, \Sigma)$) and minimize the objective in (1) with stochastic gradient descent. Note that when ℓ is the negative

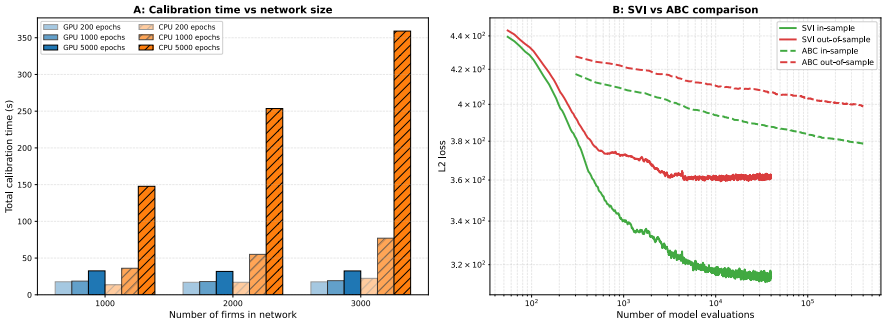

Figure 2: Two plots showing the advantages of our methodology. A: we compare the speed of SVI calibration on GPU (RTX 5090) vs CPU (Ryzen 9 9950X). B: We show how the accuracy per model evaluation decreases between SVI and ABC.

log-likelihood and D is the Kullback–Leibler divergence, GVI becomes the usual variational inference. For our purposes, we take l to simply be the L_2 loss, and D to be the Kullback–Leibler divergence. As a gradient-free baseline, we utilise a variant of Approximate Bayesian Computation (ABC), where we repeatedly sample from the prior, and keep the top k samples (for us, 100) with the lowest loss under ℓ and $\mathbf{n}$.

4 Results

To demonstrate the advantages of our methodology, we run two separate experiments, shown in Figure 2. Firstly, in Fig. 2 A, we show how our problem is naturally parallelizable by comparing the same code running on a GPU (RTX 5090) vs CPU (Ryzen 9 9950X). The benchmark used for comparison simply runs SVI for a fixed number of epochs, with different numbers of synthetic firms. The CPU-based implementation takes far longer, especially as the number of firms increase. On the other hand, the GPU-based implementation is much quicker, and doesn't increase dramatically as the number of firms increase, this is because the GPU is not using all it's parallel capacity, even at 3000 firms. As the number of firms increases, the difference between the two methods becomes only more dramatic.

In Fig. 2 B, we compare to a gradient-free baseline, showing that using a gradient based method yields huge speed-ups. We do this by taking our model, which consists of 1,000 firms, and sampling some true parameters $\mathbf{n}$ from the prior, 2 for each firm. We then generate observed data from $\mathbf{n}$, and then have the two methods train on this data, utilising the true prior. To measure potential over-fitting, we compare both the "in-sample" loss, that is the loss to the observed data, and the "out-of-sample" loss, which is the average loss to observed data re-sampled from the true prior, conditioning on the true parameters $\mathbf{n}$.

As we see from the experiment, SVI makes it possible to efficiently calibrate ABMs with 1000s of parameters, in contrast to gradient free methods like ABC. After only 300 model evaluations, SVI has a lower loss than ABC after 30,000 samples. Note that this comparison is slightly unfair, as in SVI, a model evaluation also includes a gradient pass, which adds computation that isn't measured here. However, the gradient pass takes comparable time to the forward pass, and as a result, measuring wall clock time does not meaningfully change the results.

5 Conclusion

Identifying and mitigating vulnerabilities in supply chains is crucial to supporting the well-being and prosperity of the general public. Doing this effectively requires accurate models of production networks. Agent-based models (ABMs) are a promising approach, but are hard to calibrate. By implementing a 1,000-firm production-network ABM in `JAX`, we exploit parallelisation and automatic differentiation to perform Bayesian inference orders of magnitude faster than non-differentiable alternatives. This work not only scales ABMs but also enhances their realism—for example, by enabling simulations of price dynamics, logistics, and network restructuring in response to disruptions.

References

[1] Daron Acemoglu, Vasco M. Carvalho, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. The network origins of aggregate fluctuations. *Econometrica*, 80(5):1977–2016, 2012.

[2] Robert L. Axtell and J. Doyne Farmer. Agent-based modeling in economics and finance: Past, present, and future. *Journal of Economic Literature*, 63(1):197–287, March 2025.

[3] David Rezza Baqaee. Cascading failures in production networks. *Econometrica*, 86(5):1819–1838, 2018.

[4] David Rezza Baqaee and Emmanuel Farhi. The macroeconomic impact of microeconomic shocks: Beyond hulten's theorem. *Econometrica*, 87(4):1155–1203, 2019.

[5] Jan Boelts, Michael Deistler, Manuel Gloeckler, Álvaro Tejero-Cantero, Jan-Matthis Lueckmann, Guy Moss, Peter Steinbach, Thomas Moreau, Fabio Muratore, Julia Linhart, Conor Durkan, Julius Vetter, Benjamin Kurt Miller, Maternus Herold, Abolfazl Ziaeemehr, Matthijs Pals, Theo Gruner, Sebastian Bischoff, Nastya Krouglova, Richard Gao, Janne K. Lappalainen, Bálint Mucsányi, Felix Pei, Auguste Schulz, Zinovia Stefanidi, Pedro Rodrigues, Cornelius Schröder, Faried Abu Zaid, Jonas Beck, Jaivardhan Kapoor, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. sbi reloaded: a toolkit for simulation-based inference workflows. *Journal of Open Source Software*, 10(108):7754, 2025.

[6] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[7] Oya Celasun, Aiko Mineshima, Niels-Jakob Hansen, Jing Zhou, and Mariano Spector. Supply Bottlenecks: Where, Why, How Much, and What Next? *IMF Working Papers*, 2022(031):1, 2 2022.

[8] Ayush Chopra, Alexander Rodríguez, Jayakumar Subramanian, Arnau Quera-Bofarull, Balaji Krishnamurthy, B. Aditya Prakash, and Ramesh Raskar. Differentiable Agent-based Epidemiology. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pages 1848–1857, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. event-place: London, United Kingdom.

[9] CNN Business. Federal reserve chair jerome powell: "we don't have the kind of tools to deal with supply chain problems". `http://archive.today/2025.05.07-212523/https://edition.cnn.com/business/live-news/federal-reserve-interest-rate-05-07-24`, 2025. Live transcript, May 7, 2025. Accessed: 2025-10-01.

[10] Francesco Cozzi, Marco Pangallo, Alan Perotti, André Panisson, and Corrado Monti. Learning individual behavior in agent-based models with graph diffusion networks, 2025.

[11] Joel Dyer, Patrick Cannon, J. Doyne Farmer, and Sebastian M. Schmon. Black-box bayesian inference for agent-based models. *Journal of Economic Dynamics and Control*, 161:104827, 2024.

[12] Stéphane Hallegatte. An adaptive regional input-output model and its application to the assessment of the economic cost of katrina. *Risk Anal.*, 28(3):779–799, June 2008.

[13] Stéphane Hallegatte. Modeling the role of inventories and heterogeneity in the assessment of the economic costs of natural disasters. *Risk Anal.*, 34(1):152–167, January 2014.

[14] Hiroyasu Inoue and Yasuyuki Todo. Firm-level propagation of shocks through supply-chain networks. *Nat. Sustain.*, 2(9):841–847, August 2019.

[15] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An Optimization-centric View on Bayes' Rule: Reviewing and Generalizing Variational Inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022.

[16] Francesco Lamperti, Andrea Roventini, and Amir Sani. Agent-based model calibration using machine learning surrogates, 2017.

[17] Wassily Leontief. Quantitative input and output relations in the economic system of the united states. *The Review of Economics and Statistics*, 18(3):105–125, 1936.

[18] Jr. Long, John B. and Charles I. Plosser. Real business cycles. *Journal of Political Economy*, 91(1):39–69, 1983.

[19] Corrado Monti, Marco Pangallo, Gianmarco De Francisci Morales, and Francesco Bonchi. On learning agent-based models from data. *Scientific Reports*, 13(1):9268, June 2023. Publisher: Nature Publishing Group.

[20] Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

[21] Anton Pichler, Jan Hurt, Tobias Reisch, Johannes Stangl, and Stefan Thurner. Economic impacts of a drastic gas supply shock and short-term mitigation strategies. *Journal of Economic Behavior & Organization*, 227:106750, November 2024.

[22] Anton Pichler, Marco Pangallo, R. Maria del Rio-Chanona, François Lafond, and J. Doyne Farmer. Forecasting the propagation of pandemic shocks with a dynamic input-output model. *Journal of Economic Dynamics and Control*, 144:104527, November 2022.

[23] Donovan Platt. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859, 2020.

[24] Arnau Quera-Bofarull, Nicholas Bishop, Joel Dyer, Daniel Jarne Ornia, Anisoara Calinescu, Doyne Farmer, and Michael Wooldridge. Automatic differentiation of agent-based models, 2025.

[25] I. Vernon, J. Owen, J. Aylett-Bullock, C. Cuesta-Lazaro, J. Frawley, A. Quera-Bofarull, A. Sedgewick, D. Shi, H. Truong, M. Turner, J. Walker, T. Caulfield, K. Fong, and F. Krauss. Bayesian emulation and history matching of june. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2233):20220039, 2022.