Computing Systemic Risk Measures with Graph Neural Networks

L. Gonon[1], T. Meyer-Brandis[2], N. Weber[2]

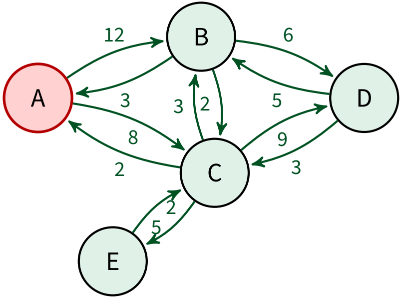
[1]University of St. Gallen
[2]LMU Munich

10th July 2026

Swissquote Workshop on Graph Learning in Financial Networks

Outline

Interbank Exposures and Systemic Risk



Nodes = banks, arrows = bilateral liabilities

Systemic risk: the failure of one institution propagates through the network and threatens a large part of the system.

- Banks are tied together by **bilateral exposures** – interbank loans, repos, derivatives.
- Each pair can owe *both ways*; if **A** cannot repay, **B** and **C** take losses and may then fail to pay **D** and **E** – a **default cascade**.
- **Example:** the 2008 collapse of *Lehman Brothers* froze interbank lending and spread losses far beyond its direct counterparties.

Question: *how should we measure systemic risk in such networks?*

Question: *how much capital should a lender of last resort reserve, and where should it be injected, to secure such a network?*

Outline

Universität St.Gallen

Systemic Risk Measures

Question: *how should we measure systemic risk in such networks?*

Widely recognized as key question in the aftermath of 2007/2008 financial crisis, see, e.g.,

- V. V. Acharya, L. H. Pedersen, T. Philippon, and M. Richardson, *Measuring systemic risk* , Rev. Financial Stud. (2017).
- T. Adrian and M. K. Brunnermeier, *CoVaR*, Amer. Econom. Rev. (2016).

Systemic risk measures I: Let $X = (X_1, \dots, X_N) \in L^0(\mathbb{R}^N)$ represent random risk factors of a system of N institutions.

- Aggregation function $\Lambda : \mathbb{R}^N \to \mathbb{R}$
- Univariate risk measure $\eta : L^0(\mathbb{R}) \to \mathbb{R}$

Chen et al.[1] approach: systemic risk measure **aggregates first, then adds capital**:

$$\rho(X) = \eta(\Lambda(X)), \quad \rho(X) = \inf\{m \in \mathbb{R} \mid \Lambda(X) + m \in \mathbb{A}\},$$

assuming η is cash-invariant and $\mathbb{A} = \{Z \mid \eta(Z) \leq 0\}$ is the acceptance set of η.

[1] C. Chen, G. Iyengar, and C. C. Moallemi, *An axiomatic approach to systemic risk*, Manag. Sci. (2013).

Systemic Risk Measures

Question: *how should we measure systemic risk in such networks?*

Systemic risk measures II: *Allocate before aggregation* ("prevent contagion *before* it occurs")

- Z. Feinstein, B. Rudloff, and S. Weber, *Measures of systemic risk* , SIAM J. Financial Math. (2017)
- F. Biagini, J.-P. Fouque, M. Frittelli, and T. Meyer-Brandis, *A unified approach to systemic risk measures via acceptance sets*, Math. Finance (2019)
- F. Biagini, J.-P. Fouque, M. Frittelli, and T. Meyer-Brandis, *On fairness of systemic risk measures*, Finance Stoch. (2020)

Systemic risk measure is defined as

$$\rho(X) = \inf_{Y \in \mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \,\middle|\, \Lambda(X+Y) \in \mathbb{A} \right\},$$

with available allocations $\mathcal{C}$ a subset of

$$\mathcal{C}_{\mathbb{R}} = \left\{ Y \in L^0(\mathbb{R}^N) \,\middle|\, \sum_{i=1}^{N} Y_i \text{ is constant a.s.} \right\}.$$

Optimal Bailout Capital

Question: *how much capital should a lender of last resort reserve, and where should it be injected, to secure such a network?*

Goal: for risk-level $b \geq 0$, compute

$$\rho_b(X) = \inf_{Y \in \mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \,\middle|\, \eta\left(\Lambda(X+Y)\right) \leq b \right\},$$

the smallest amount of bailout capital required to make aggregate risk acceptable and a minimizing Y^* (optimal allocation).

Lender of last resort reserves *fixed* amount of bailout capital, but waits with distributing to each institution to see which scenario is realized.

Note: this covers both “Systemic risk measures I” and “Systemic risk measures II”.

This talk: *graph neural network-based algorithm for computing $\rho_b(X)$ and optimal allocation Y^* for financial networks.*

Outline

Optimal Bailout Capital

Universität St.Gallen

Goal: compute optimal bailout capital

$$\rho_b(X) = \inf_{Y \in \mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \,\middle|\, \eta\left(\Lambda(X+Y)\right) \le b \right\},$$

and optimal allocation Y^*.

Ingredients:

- Modelling financial networks ($\leftrightarrow$ choosing X)
- Systemic risk aggregation ($\leftrightarrow$ choosing Λ)
- Iterative optimization algorithm

Financial networks

Universität St.Gallen

We model a financial network of N institutions and their interbank liabilities as follows:

1. The institutions' **assets** $a = (a_1, \ldots, a_N) \in (0, \infty)^N$
2. The interbank **liability matrix** $\ell \in [0, \infty)^{N \times N}$, where $\ell_{ij} > 0$ indicates a liability of size ℓ_{ij} from bank i towards bank j. Note $\mathrm{diag}(\ell) = (\ell_{11}, \ldots, \ell_{NN}) = (0, \ldots, 0)$.

One-to-one correspondence $g = (a, \ell)$ to **directed** graphs with **node features** and **edge features**.

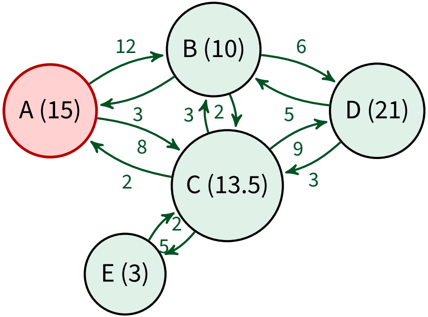
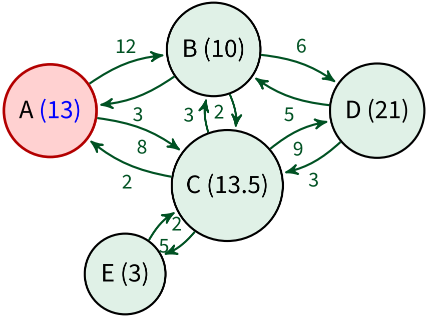


Figure: Example network

Contagion in financial networks

Assets and incoming cash flows of an institution may not be able to meet the full amount of outgoing liabilities.



$\rightarrow$ other participants may not be able to meet their liabilities due to the lack of incoming cash flows

$\rightarrow$ chain reaction of defaults propagated through the network.

Eisenberg-Noe Clearing Mechanism

Universität St.Gallen

- A bank's assets and incoming payments may not cover its outgoing liabilities. If it cannot pay, its creditors receive less and may default in turn: a **contagious cascade** of defaults along the network.
- Eisenberg and Noe[2]: find the payments that are mutually consistent in equilibrium – what everyone will pay in the end. Three principles:
  - **Limited liability**: a bank never pays more than the funds it has.
  - **Proportionality**: in default, creditors are repaid proportionally.
  - **Absolute priority**: in default, *all* available funds are used for repayment.
- The resulting consistent payment profile is the **clearing vector**; it tells us how much value is actually available at each node, hence how large the systemic loss is.

[2] L. Eisenberg and T. H. Noe, *Systemic risk in financial systems*, Manage. Sci. (2001)

Eisenberg-Noe Clearing Mechanism

For a financial network $(a, \ell) \in (0, \infty)^N \times [0, \infty)^{N \times N}$, define total and relative liabilities

$$\bar{\ell}_i := \sum_{j=1}^{N} \ell_{ij}, \qquad \pi_{ij} := \begin{cases} \ell_{ij}/\bar{\ell}_i & \text{if } \bar{\ell}_i > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose each institution k is able to make payment p_k, then the incoming payments of institution i will be $\sum_{k=1}^{N} \pi_{ki} p_k = (\pi^\top p)_i$.

The **clearing map** $\Phi(\cdot, a, \pi, \bar{\ell}) : \mathbb{R}^N \to \mathbb{R}^N$ measures the debt each node can repay if the others repay p:

$$\Phi(p, a, \pi, \bar{\ell}) := \Big(\underbrace{\pi^\top p}_{\text{incoming}} + \underbrace{a}_{\text{assets}} \Big) \wedge \underbrace{\bar{\ell}}_{\text{owed}}.$$

- A **clearing vector** is a fixed point: $\mathrm{CV}(a, \ell) = \Phi\big(\mathrm{CV}(a, \ell), a, \pi, \bar{\ell}\big)$.
- Properties of $\Phi \Rightarrow$ there exists a **unique** clearing vector.
- Computable by iterating Φ from 0 (or by solving finitely many linear systems): $\mathrm{CV}(a, \ell) = \lim_{n \to \infty} \Phi^{\circ n}(0, a, \pi, \bar{\ell})$.

Aggregation Functions

Aggregated systemic loss: our key example is the *total payment shortfall*

$$\Lambda(a, \ell) = \sum_{i=1}^{N} \bar{\ell}_i - \mathrm{CV}_i(a, \ell)$$

Each term $\bar{\ell}_i - \mathrm{CV}_i$ is the liabilities of node i that are not paid; the sum is the **total unpaid debt** in the network.

This choice of Λ is a key example satisfying our assumption:

Assumption

The aggregation function $\Lambda : (0, \infty)^N \times [0, \infty)^{N \times N} \to \mathbb{R}$ *satisfies*

- Λ *is convex and decreasing in its first component;*
- *for every* $u < \infty$ *there is a deterministic* $y_u^* \in \mathbb{R}_+^N$ *with* $\Lambda(\cdot + y_u^*, \ell) = 0$ *for all* $\ell \leq u$*;*
- *there exist* $c_1, c_2 \geq 0$ *with* $\Lambda(\cdot, \ell) \leq c_1 |\ell|_1 + c_2$.

Other Admissible Aggregation Functions

With relative liabilities π and a designated "society" node j^*:

- **Shortfall to society**: $\Lambda(a,\ell) = \sum_{i \neq j^*} \pi_{ij^*} \left(\bar{\ell}_i - \mathrm{CV}_i(a,\ell) \right)$.
- **Risk-weighted shortfalls** with coefficients $c, c_j \geq 0$:

  $$\Lambda(a,\ell) = c \sum_{j=1}^{N} c_j \sum_{i=1}^{N} \pi_{ij} \left(\bar{\ell}_i - \mathrm{CV}_i(a,\ell) \right).$$
- **Risk-averse** variants applying increasing convex u (e.g. $u(x) = e^{\gamma x}$) to nodal or total shortfall, e.g.,

  $$\Lambda(a,\ell) = \sum_{j=1}^{N} u_j \left(\sum_{i=1}^{N} \pi_{i,j} \left(\bar{\ell}_i - \mathrm{CV}_i(a,\ell) \right) \right),$$

  assuming $L \in L^\infty$ below.

Optimal Bailout Capital

Goal: compute optimal bailout capital

$$\rho_b(X) = \inf_{Y \in \mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \,\middle|\, \eta\left(\Lambda(X+Y)\right) \leq b \right\},$$

and optimal allocation Y^*.

Ingredients:

- Modelling financial networks ($\leftrightarrow$ choosing X) ($\checkmark$)
- Systemic risk aggregation ($\leftrightarrow$ choosing Λ) $\checkmark$
- Iterative optimization algorithm

Random Financial Network

- Consider **random** financial networks
- Allows to incorporate uncertainty about future state of financial network (or partially known interbank data).
- Probability space $(\Omega, \mathcal{F}, \mathbb{P})$
- $N \in \mathbb{N}$ institutions.
- Network state modelled as graph-valued random variable $G = (A, L)$:
  - external assets $A = (A_1, \ldots, A_N) \in L^0(\mathbb{R}_+^N)$,
  - interbank liability matrix $L = (L_{ij})_{i,j} \in L^1(\mathbb{R}_+^{N \times N})$, $\mathrm{diag}(L) = 0$.
- Set of all (integrable) stochastic financial networks:

$$\mathcal{G} := \left\{ (A, L) \,\middle|\, A \in L^0(\mathbb{R}^N_{++}),\ L \in L^1(\mathbb{R}_+^{N \times N}),\ \mathrm{diag}(L) = 0 \right\}.$$

- We work with strictly positive assets $A \in \mathbb{R}^N_{++}$ (ensures a unique clearing vector).

Systemic Risk Measure with Random Allocations

- With an aggregation function $\Lambda : \mathbb{R}^N_{++} \times \mathbb{R}^{N\times N}_+ \to \mathbb{R}$ and univariate η, our goal is to compute the minimal bailout capital:

$$\rho_b(A, L) := \inf_{Y\in\mathcal{C}} \left\{ \sum_{n=1}^{N} Y_n \;\middle|\; \eta\big(\Lambda(A + Y, L)\big) \leq b \right\},$$

  with admissible *nonnegative* allocations whose components sum to a constant a.s.,

$$\mathcal{C} = \left\{ Y \in L^0(\mathbb{R}^N_+) \;\middle|\; \sum_{i=1}^{N} Y_i \text{ is constant a.s.} \right\}.$$

- Reserve a fixed amount of bailout capital *today*; **allocate it scenario-dependently** once ω is realized.
- Extends Biagini et al. [1] ($\mathbb{R}^N \to$ graph-valued) and Feinstein et al. [2] (deterministic $\to$ random allocations).
- $Y \geq 0$: no cross-subsidization

Optimal Bailout Capital

Goal: compute optimal bailout capital

$$\rho_b(A, L) := \inf_{Y \in \mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \;\middle|\; \eta\big(\Lambda(A + Y, L)\big) \le b \right\},$$

and optimal allocation Y^*.

Ingredients:

- Modelling financial networks $\checkmark$
- Systemic risk aggregation ($\leftrightarrow$ choosing Λ) $\checkmark$
- Iterative optimization algorithm

Computing Optimal Bailout Capital: Overview

Step 1: rewrite

$$\rho_b(A,L) = \inf_{Y\in\mathcal{C}} \left\{ \sum_{i=1}^{N} Y_i \,\middle|\, \eta(\Lambda(A+Y,L)) \le b \right\} = \inf\{c \in \mathbb{R}_+ | \rho^I_c(G) \le b\},$$

for $b > 0$ and with $\rho^I_c(G)$ the **inner risk** at fixed capital c:

$$\rho^I_c(G) = \inf_{Y\in\mathcal{C}^c} \eta(\Lambda(A+Y,L))$$

$$\mathcal{C}^c = \{Y \in L^0(\mathbb{R}^N_+) | \sum_n Y_n = c\}$$

Step 2: show existence of an **optimal allocation function** $H^c : (0,\infty)^N \times [0,\infty)^{N\times N} \to [0,\infty)^N$ such that $Y^* = H^c(A,L) \in \mathcal{C}^c$ is an optimal allocation for capital level c:

$$\rho^I_c(G) = \eta(\Lambda(A+Y^*,L))$$

Computing Optimal Bailout Capital: Overview

Step 3: approximate the optimal allocation function H^c by a neural network ϕ_θ^c

- Naive: feedforward neural network
- Refined: graph neural network

Step 4: use *probabilistic bisection search* to iteratively:

- improve neural network parameters θ for given c
- improve c based on risk estimate with current θ

Assumptions

Assumption

1. *The univariate risk measure η is monotone increasing, convex and normalized ($\eta(0) = 0$).*
2. *Assets are strictly positive: $A \in \mathbb{R}^N_{++}$.*
3. *The liability matrix is integrable: $L \in L^1(\mathbb{R}^{N\times N}_+)$.*
4. *The aggregation function $\Lambda : \mathbb{R}^N_{++} \times \mathbb{R}^{N\times N}_+ \to \mathbb{R}$ satisfies*
    - *Λ is convex and decreasing in its first component;*
    - *for every $u < \infty$ there is a deterministic $y^*_u \in \mathbb{R}^N_+$ with $\Lambda(\cdot + y^*_u, \ell) = 0$ for all $\ell \leq u$;*
    - *there exist $c_1, c_2 \geq 0$ with $\Lambda(\cdot, \ell) \leq c_1|\ell|_1 + c_2$.*

- The above examples satisfy all hypotheses.
- The hypotheses on η imply that η is continuous on L^1.
- L^∞ **variant:** alternatively, we could require $L \in L^\infty$ and drop the growth bound on Λ.

Main Theoretical Results

Theorem (G., Meyer-Brandis, Weber)

Let $(A, L) \in \mathcal{G}$ and $b > 0$. Then:

1. $\rho_b(A, L)$ *is well-defined:* $-\infty < \rho_b(A, L) < +\infty$.
2. *An optimal allocation exists, i.e., the infimum in* $\rho_b(A, L)$ *is* ***attained***.
3. *For each optimal* Y^* *the risk equals the threshold exactly:* $\eta\big(\Lambda(A + Y^*, L)\big) = b$.
4. *The set of all optimal allocations is* ***convex***.

- Optimal allocations need not be unique.
- Proofs rely on **Komlós' Theorem** and continuity/convexity of η and Λ.

Measurable Optimal Allocations

- **Inner risk** at fixed capital c: $\rho^I_c(G) = \inf_{Y \in \mathcal{C}^c} \eta(\Lambda(A+Y, L))$, where $\mathcal{C}^c = \{Y \in L^0(\mathbb{R}^N_+) : \sum_n Y_n = c\}$.
- **Outer risk**: $\rho^O_b(G) = \inf\{c \in \mathbb{R}_+ : \rho^I_c(G) \leq b\}$.

Proposition

$\rho_b(G) = \rho^O_b(G)$, and both infima are attained.

Theorem

*If Λ is measurable, then for every $c > 0$ there exists a **measurable** function*

$$H^c : \mathbb{R}^N_{++} \times \mathbb{R}^{N \times N}_+ \to \mathbb{R}^N_+$$

such that $Y^c = H^c(A, L) \in \mathcal{C}^c$ is an optimal allocation for capital level c, that is,

$$\rho^I_c(G) = \eta(\Lambda(A + Y^c, L)).$$

- Measurability $\Rightarrow$ H^c can be approximated by neural networks.

Iterative Algorithm

Universität St.Gallen

Goal: cheapest allocation Y securing the future network. Write $H^c = c\,\varphi^c$, approximate $\varphi^c \approx \phi^\theta$ (softmax output, sums to one).

Guess initial bailout capital $c \in \mathbb{R}_+$
for $j = 1, \dots, N_{\text{epochs}}$ **do**
    Sample realizations $(a^i, \ell^i)_{i \in N_{\text{sample}}}$ of (A, L)
    Approximate optimal allocation $H^c(A, L) \approx c\,\phi^\theta(A, L)$:
    Estimate the risk $\eta(\Lambda(A + c\,\phi^\theta(A, L), L))$
    for $k = 1, \dots, N_{\text{steps}}$ **do**
        update θ by SGD on the empirical aggregated risk
    end for
    If risk $> b$, increase c; otherwise decrease c (use *probabilistic bisection*)
end for
Output: c, ϕ^θ

Outline

Approximate Optimal Bailout Function

Naive approach:

- Use feedforward neural networks (FNNs)
- Approximate $H^c(a, \ell) \approx \mathrm{FNN}(\mathrm{vec}(a, \ell))$.
- FNNs are **universal approximators** of measurable functions (in probability).

Issues:

- Observe: H^c is **permutation-equivariant**

$$\sigma(H^c(a, \ell)) = H^c(\sigma(a, \ell)),$$

  for permutations σ of $\{1, \ldots, N\}$; relabelling nodes permutes the optimal allocation accordingly.
- FNNs are *not* permutation-equivariant in general:

$$\mathrm{FNN}(\sigma(x)) \neq \sigma(\mathrm{FNN}(x)).$$

  Equivariance must be learned from data, this makes the learning task harder.
- Feeding the full liability matrix $\mathrm{vec}(a, \ell)$ is computationally expensive: the first layer alone connects $\sim N + N^2$ inputs.

Graphs and Permutation Equivariance

- For our financial networks, nodes have **no natural order**: the same network has many valid representations.
- For a permutation σ of $\{1, \ldots, N\}$, the permuted graph $\sigma(a, \ell) = (\sigma(a), \sigma(\ell))$ is

$$\sigma(a)_i = a_{\sigma^{-1}(i)}, \qquad \sigma(\ell)_{i,j} = \ell_{\sigma^{-1}(i),\sigma^{-1}(j)}.$$

- A node-labeling function $\tau : \mathbb{R}^N \times \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times l}$ is **permutation-equivariant** if

$$\tau(\sigma(a, \ell)) = \sigma(\tau(a, \ell))$$

  for all permutations of of $\{1, \ldots, N\}$.
- The optimal allocation function H^c is permutation-equivariant $\Rightarrow$ we want architectures respecting this *by design*.

Graph Neural Networks (GNNs)

- GNNs operate directly on graph data and have been very successful in numerous applications.
- We consider **message-passing GNNs**
- We introduce a general GNN architecture, **extended permutation-equivariant neural networks (XPENNs)**, that subsume many commonly used architectures.
- GNNs are **permutation-equivariant** by construction.
- *Classical* message-passing GNNs are **not universal** (bounded by the Weisfeiler–Lehman test / tree-like equivalence).
- XPENNs **are universal** (see below).

Message-passing GNNs

A message-passing layer **aggregates messages from a node's neighborhood** and updates a *representation* for each node. Let $N(i) = \{j \in \{1, \dots, N\} \setminus \{i\} | \ell_{ji} \neq 0\}$ be the neighborhood of i.

Definition

Let $h_i^{(l)}$ be the representation of node $i \in \{1, \dots, N\}$ in iteration $l \in \{0, \dots, L\}$. Set

$$
\begin{aligned}
h_i^{(0)} &= a_i, \\
h_i^{(l)} &= f_{act}(f_{com}^{(l)}(h_i^{(l-1)}, f_{ag}^{(l)}(\{(h_j^{(l-1)}, \ell_{ji}) \mid j \in N(i)\}))), \qquad \text{for } l=1,\dots,L-1\,, \\
h_i^{(L)} &= f_{com}^{(L)}(h_i^{(L-1)}, f_{ag}^{(L)}(\{(h_j^{(L-1)}, \ell_{ji}) \mid j \in N(i)\})),
\end{aligned}
$$

with $f_{act} : \mathbb{R} \to \mathbb{R}$ is applied componentwise and FNN-type functions $f_{ag}^{(l)}, f_{com}^{(l)}$. Then f with

$$
f(a, \ell)_i = h_i^{(L)}
$$

is called a graph neural network with L rounds of message passing.

Characterization of Permutation-Equivariant Functions

Universität St.Gallen

Proposition (extends [3])

*On a graph domain $\mathcal{D}$ where the node features provide finite unique node features, a measurable $\tau : \mathcal{D} \to \mathbb{R}^{N \times l}$ is permutation-equivariant **if and only if** there exist measurable $\rho, \alpha, \psi, \varphi$ such that for all (a, ℓ) and $k = 1, \ldots, N$,*

$$\tau(a, \ell)_k = \rho\left(a_k, \underbrace{\sum_{j \neq k} \psi(a_k, \ell_{kj}, \ell_{jk}, a_j),}_{\text{node-specific signal}} \underbrace{\sum_{i=1}^{N} \alpha\left(a_i, \sum_{j \neq i} \varphi(a_i, \ell_{ij}, a_j)\right)}_{\text{graph-wide summary}} \right).$$

- Constructive proof: φ, α encode the entire graph into the second argument; ρ then "reads off" τ.
- **Motivates a neural architecture**: replace $\rho, \alpha, \psi, \varphi$ by FNNs.

Herzig et al. [3] *Mapping images to scene graphs with permutation-invariant structured prediction.* Neurips 2018.

(Extended) Permutation-Equivariant Neural Networks

For FNNs $\hat{\rho}, \hat{\alpha}, \hat{\varphi}$, the **PENN** aggregates graph information *without* iterated message passing [3]:

$$\mathrm{PENN}(a,\ell)_k = \hat{\rho}\left(a_k, \sum_{i=1}^{N} \hat{\alpha}\Big(a_i, \sum_{j\neq i} \hat{\varphi}(a_i, \ell_{ij}, a_j)\Big)\right).$$

The **XPENN** adds a node-specific component (both in- and out-edges), recovering one round of message passing:

$$\mathrm{XPENN}(a,\ell)_k = \hat{\rho}\left(a_k, \sum_{j\neq k} \hat{\psi}(a_k, \ell_{kj}, \ell_{jk}, a_j), \sum_{i=1}^{N} \hat{\alpha}\Big(a_i, \sum_{j\neq i} \hat{\varphi}(a_i, \ell_{ij}, a_j)\Big)\right).$$

- Both are **permutation-equivariant**; every PENN is a special XPENN ($\hat{\psi} \equiv 0$).
- The extra component prevents node-specific information from being “lost” in the outer sum.

Spoiler: Universal Approximation for (X)PENNs

Universität St.Gallen

Theorem

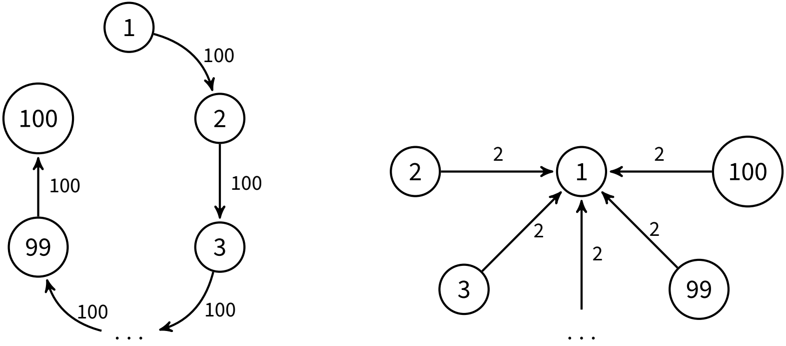
Let ρ, α, φ (ψ) be Borel measurable and g the corresponding (X)PENN-type composition. For any probability measure μ on $\mathbb{R}^{N\times d_1} \times \mathbb{R}^{N\times N\times d_2}$, any $p \in [1, \infty]$, and any $\varepsilon, \bar{\varepsilon} > 0$, there exist FNNs $\hat{\rho}, \hat{\alpha}, \hat{\varphi}$ ($\hat{\psi}$) so that the resulting $\hat{g}$ satisfies

$$\mu\big(\{(x,y) : |g(x,y) - \hat{g}(x,y)|_p \geq \bar{\varepsilon}\}\big) \leq \varepsilon.$$

- Together with the previous characterization, this proves: **(X)PENNs approximate any permutation-equivariant node-labeling function in probability**
- In particular, this applies to the optimal allocation function H^c.
- This is a property that classical message-passing GNNs do *not* enjoy.

Outline

Experiment 1: Stylized Cascade & Star Networks



- For each $N \in \{10, 20, 50, 100\}$: dataset of $2N$ networks as in Figure; all external assets equal 1; $\eta = \mathbb{E}[\cdot]$, $\Lambda =$ *total payment shortfall (TPS)*.
- Each network can be **fully secured** (TPS $= 0$) with capital $c = N - 1$:
  - cascade $\to$ inject all capital at the start
  - star $\to$ one unit to each leaf.

Experiment 1 – Approximation

N	Model	Epoch	Avg. TPS	Capital	Time (mm:ss)
10	None	–	27.00	0.00	–
10	GNN	1000	0.00	9.00	01:09
10	XPENN	1000	0.00	9.00	00:47
10	FNN(L)	1000	0.23	9.00	00:23
20	None	–	104.50	0.00	–
20	GNN	1000	0.00	19.00	02:45
20	XPENN	1000	0.00	19.00	01:52
20	FNN(L)	1000	2.68	19.00	00:59
50	None	–	637.00	0.00	–
50	GNN	1000	0.00	49.00	08:49
50	XPENN	1000	0.00	49.00	07:42
50	FNN(L)	1000	49.78	49.00	04:04
100	None	–	2524.50	0.00	–
100	GNN	1000	0.00	99.00	24:02
100	XPENN	1000	0.00	99.00	21:40
100	FNN(L)	1000	1200.08	99.00	21:38

Table: GNN and XPENN reach perfect bailout; FNN(L) does not, and its cost/runtime scale poorly in N.

Experiment 1 – Generalization

Universität St.Gallen

N	Model	Ep.	Train	Test	Cap.	Time
10	None	–	28.20	23.40	0.00	–
10	GNN	10	0.49	0.47	9.00	00:01
10	GNN	350	0.00	0.00	9.00	00:18
10	XPENN	10	11.15	8.47	9.00	00:01
10	XPENN	30	0.46	0.34	9.00	00:02
10	XPENN	270	0.00	0.00	9.00	00:08
10	FNN(L)	10	11.34	10.49	9.00	00:01
10	FNN(L)	1000	0.69	12.55	9.00	00:17
20	None	–	104.50	104.50	0.00	–
20	GNN	10	0.07	0.07	19.00	00:01
20	GNN	110	0.00	0.00	19.00	00:13
20	XPENN	10	38.75	38.75	19.00	00:01
20	XPENN	30	0.40	0.40	19.00	00:02
20	XPENN	230	0.00	0.00	19.00	00:20
20	FNN(L)	10	46.14	48.30	19.00	00:01
20	FNN(L)	1000	2.16	49.96	19.00	00:44

N	Model	Ep.	Train	Test	Cap.	Time
50	None	–	644.84	613.48	0.00	–
50	GNN	10	325.94	308.79	49.00	00:04
50	GNN	40	0.00	0.00	49.00	00:14
50	XPENN	10	11.19	10.60	49.00	00:04
50	XPENN	20	1.17	1.11	49.00	00:07
50	XPENN	150	0.00	0.00	49.00	00:52
50	FNN(L)	10	305.02	299.80	49.00	00:01
50	FNN(L)	1000	23.10	314.88	49.00	03:28
100	None	–	2524.50	2524.50	0.00	–
100	GNN	10	49.65	49.65	99.00	00:09
100	GNN	110	0.00	0.00	99.00	01:53
100	XPENN	10	7.49	7.49	99.00	00:09
100	XPENN	100	0.00	0.00	99.00	01:39
100	FNN(L)	10	1227.93	1246.98	99.00	00:11
100	FNN(L)	1000	455.16	1333.04	99.00	17:11

Split into 75% training data and 25% test data. FNN(L) overfits: low train risk but high test risk. GNN/XPENN generalize perfectly – relabeling nodes does not change the messages they receive.

Experiment 1 – Note on the standard PENN

- In this stylized data, the **standard PENN** struggles – even with node IDs.
- Two reasons:
  - it aggregates the “wrong” information (φ uses *outgoing* edges; here *incoming* liabilities matter);
  - node-specific information gets “lost” in the outer sum $\sum_i \alpha(\cdot)$ and is hard to recover, especially with inhomogeneous edge weights.
- The **XPENN** resolves both: it always considers in- and out-edges and carries a node-specific component.
- On the more realistic networks below, the standard PENN works very well too.

Experiments 2–4: Synthetic Network Models

- **Erdős–Rényi (ER):** homogeneous, $N = 100$, independent Beta-distributed assets, edges of size 1 with $p = 0.4$.
- **Core–Periphery (CP):** inhomogeneous, 10 large $+$ 90 small banks; group-dependent edge probabilities and sizes; correlated assets (Gaussian copula). Adapted from Feinstein et al. [2].
- **Core–Periphery fixed (CPf):** as CP but the **liability matrix is sampled once** and fixed; only assets vary.

Models. (Classical SAGEConv) GNN, PENN, XPENN; FNN, FNN(L); and benchmarks: Linear, Uniform, Default, Level-1 (first-round defaults), None.

Experiment 2 – Allocating Fixed Bailout Capital ($c = 50$)

Model	ER Val.	ER Test	CP Val.	CP Test	CPf Val.	CPf Test
None	261.64	262.08	235.56	238.34	296.30	298.50
Uniform	221.22	221.71	201.45	203.57	253.98	255.73
Level-1	174.32	174.78	158.04	160.03	208.57	210.36
GNN	173.71	174.23	153.15	155.17	**203.01**	**204.85**
PENN	173.71	174.24	153.05	155.09	204.12	205.92
XPENN	**172.74**	**173.26**	**150.21**	**152.14**	203.57	205.40
FNN	221.22	221.71	201.07	203.20	203.76	205.59
FNN(L)	221.22	221.72	201.13	203.22	204.37	206.18
Linear	178.66	179.21	159.80	161.84	210.40	212.35

Table: Inner risk at fixed capital $c = 50$ (lower is better). XPENN best on ER/CP; on CPf all flexible models are close.

- ER/CP: only equivariant models (GNN, PENN, XPENN) extract the network structure; FNN/FNN(L) $\approx$ Uniform.
- CPf (fixed L): FNN/FNN(L) work too – the liability dependence is constant and learnable.

Experiment 3 – Computing the Systemic Risk ($b = 100$)

Universität St.Gallen

Model	ER: Capital	CP: Capital	CPf: Capital
Uniform	302.91	303.14	447.88
Default	147.61	134.11	220.39
Level-1	108.54	107.55	166.03
GNN	96.03	87.47	115.72
PENN	96.02	87.05	115.09
XPENN	**95.83**	**86.67**	**114.72**
FNN	303.08	303.52	116.55
FNN(L)	303.24	303.25	116.80
Linear	113.77	113.10	139.11

Table: Smallest bailout capital making the network acceptable ($\eta = \mathbb{E}[\cdot]$, $\Lambda =$ TPS). Lower is better.

- Consistent with Experiment 2: equivariant models need the **least capital**.
- Scenario-dependent allocation (Level-1, Linear, ...) improves upon Constant/Uniform.
- Effectively processing L (GNN/PENN/XPENN) improves upon Linear/Level-1. FNN/FNN(L) fail with stochastic L.

Outline

Conclusion

- We extend systemic risk measures with random allocations from $\mathbb{R}^N$-valued to graph-valued networks (stochastic assets *and* liability matrix).
- Under reasonable assumptions (incl. Eisenberg–Noe based aggregation): optimal random allocations exist, and a reformulation yields an iterative algorithm.
- We characterize permutation-equivariant node-labeling functions and propose **(X)PENNs** with a universal approximation guarantee.
- Numerical experiments:
  - when the optimum is known, GNN and (X)PENN find it
  - GNNs and XPENNs succesfully compute optimal bailout capital
  - permutation equivariance is key to extracting structure from L
  - FNNs work only when L is *fixed*; equivariant models work in both regimes
- **Outlook:**
  - richer contagion (e.g. fire sales)
  - PENNs with random features (universality & rates) $\rightarrow$ **next talk**

References I

[1] F. Biagini, J.-P. Fouque, M. Frittelli, and T. Meyer-Brandis. A unified approach to systemic risk measures via acceptance sets. *Mathematical Finance*, 29(1):329–367, 2019. doi: https://doi.org/10.1111/mafi.12170. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/mafi.12170`.

[2] Z. Feinstein, B. Rudloff, and S. Weber. Measures of systemic risk. *SIAM Journal on Financial Mathematics*, 8(1):672–708, 2017.

[3] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *Advances in Neural Information Processing Systems*, 31, 2018.

[4] N. Keriven and G. Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

References II

[6] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Thank you for your attention!

Questions?