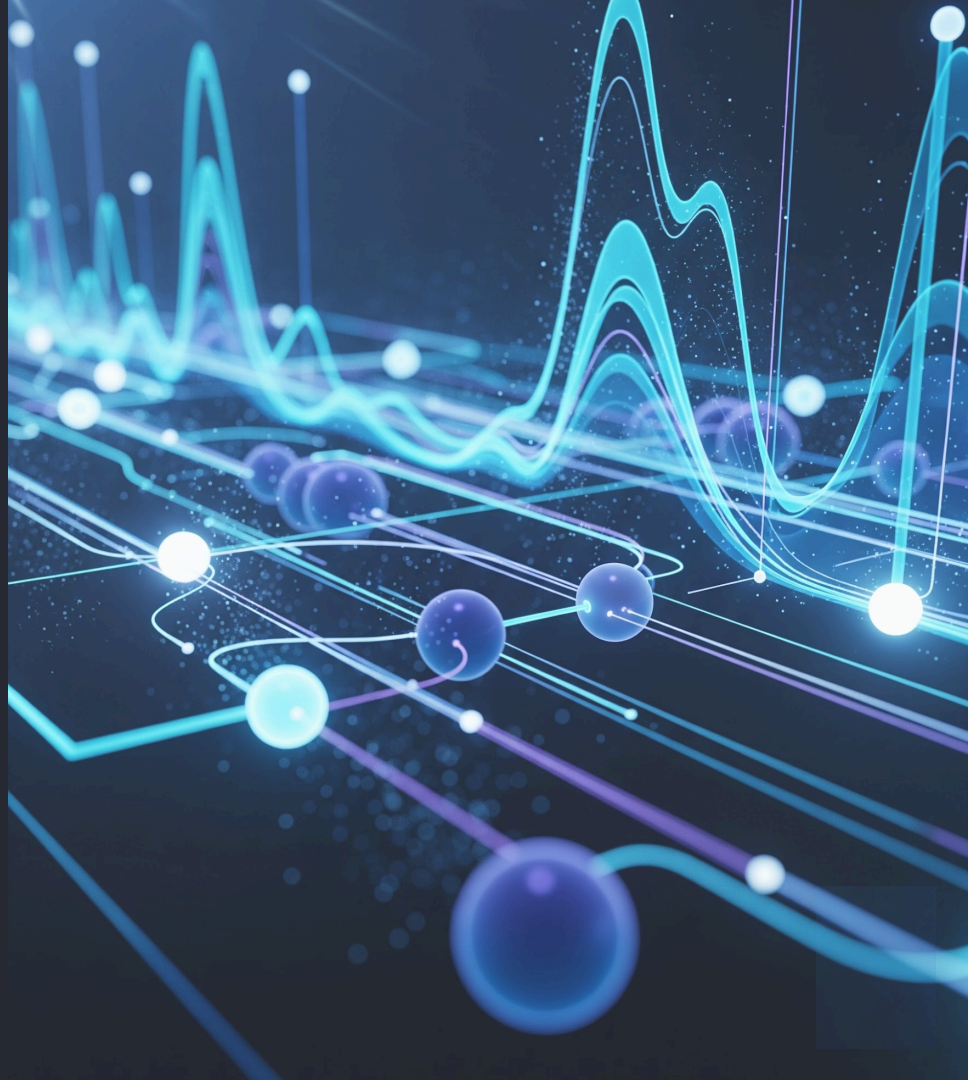
EPFL WORKSHOP ON GRAPH LEARNING IN FINANCIAL NETWORKS

Graph Machine Learning for *Dynamic Financial Networks*

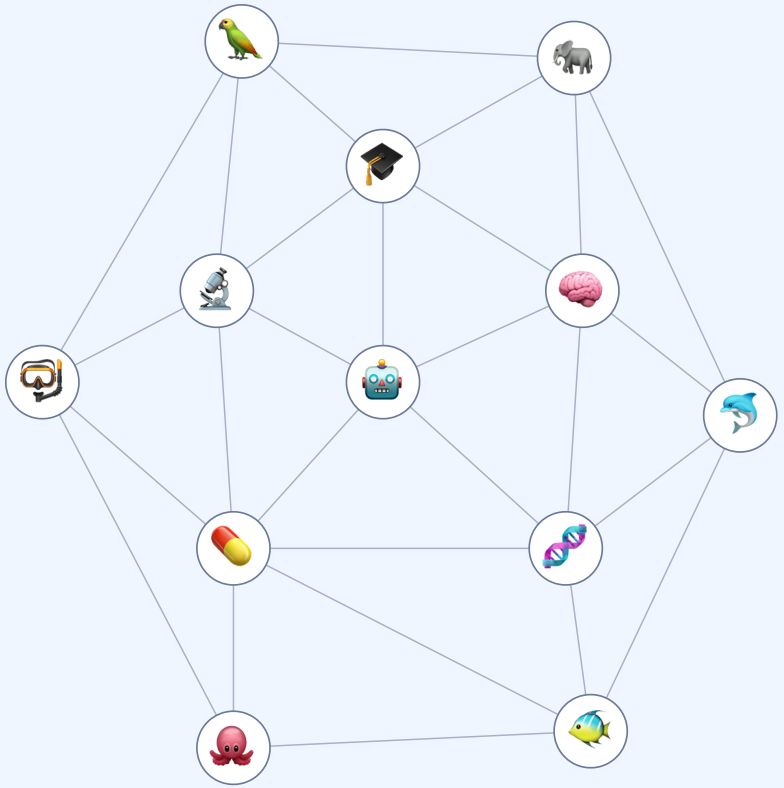
From Foundations to the Lightning Network

10/07/2026



A Bit About Myself

- 🎓 Bachelor and Master in **Computer Science** at Imperial College and Cambridge (2015-2019)
- 🤖 **PhD on Machine Learning on Graphs** between Twitter and Imperial College (2019-2023)
- 🧬 ML Researcher on **generative models for structural biology and drug discovery** at Vant AI (2024-2025)
- 🐬 PostDoc on **ML for decoding non-human communication** @ Sapienza (2026 - Current)
- **ML Advisor at Amboss Technologies**, working on ML for the Lightning Network (2024-Current)

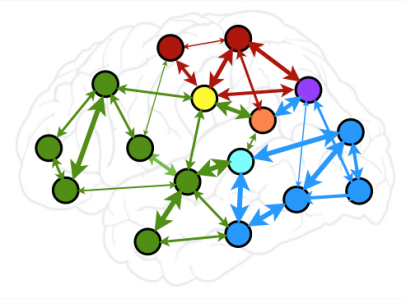


SECTION 01

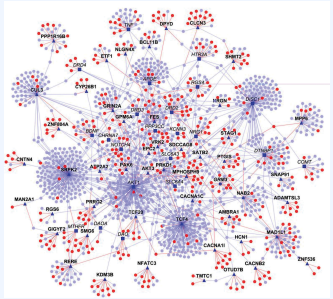
Why Should We Care About ML on Graphs?

Networks are everywhere

And graphs are a great way to model them

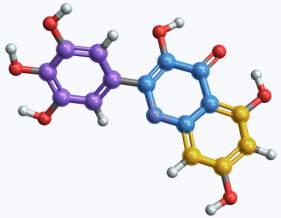


Functional Networks



Interaction Networks

Social Networks

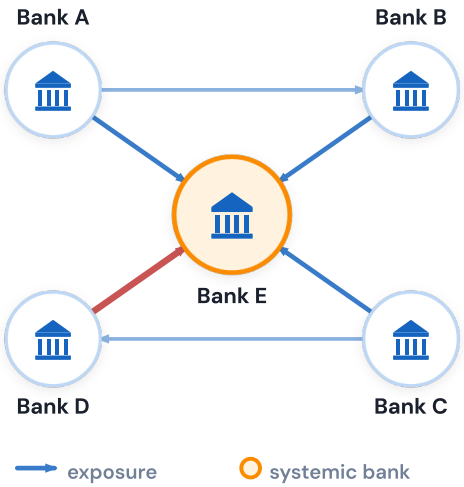


Molecules

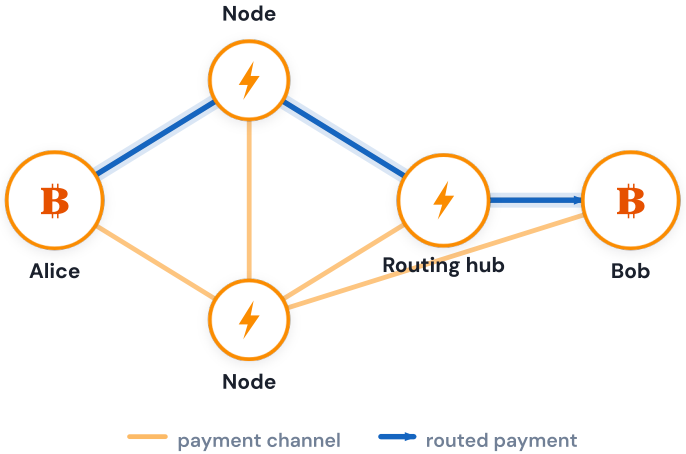
Networks are everywhere

Also in finance

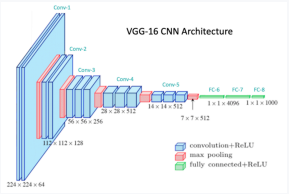
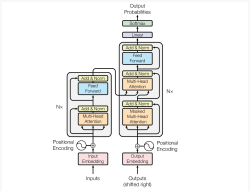
Interbank exposure network



Bitcoin Lightning network



What model should we use for graphs?

MODALITY	DATA	ARCHITECTURE
Images		
Text	`The` `cat` `sat` `on` `the` `mat`	
Graphs		?

SECTION 02

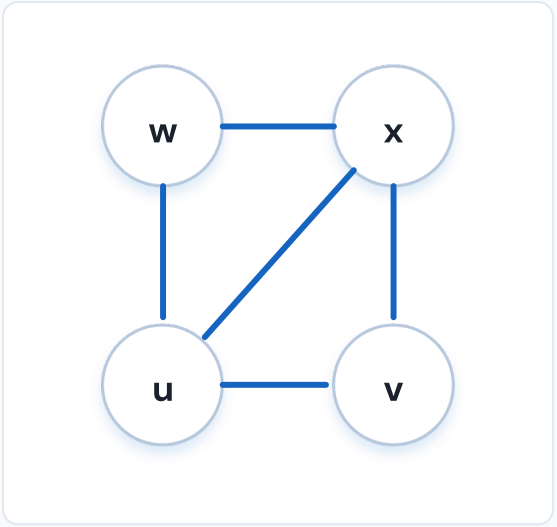
(Static) Graphs and Graph Tasks

Graphs: nodes, edges, and features

$$G = (V, E)$$

V = nodes

E = edges



$$A \in \mathbb{R}^{n \times n}$$

A = Adjacency Matrix

	u	v	w	x
u	0	1	1	1
v	1	0	0	1
w	1	0	0	1
x	1	1	1	0

$$X \in \mathbb{R}^{n \times d}$$

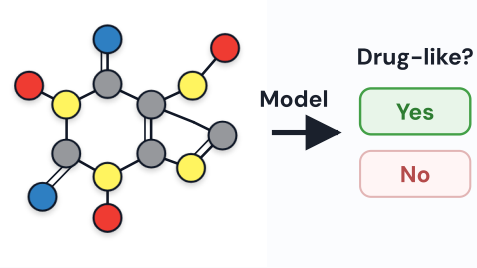
X = node features

	f1	f2	f3
u	0.8	0.4	1.7
v	0.2	0.9	1.4
w	0.6	0.1	1.9
x	0.4	0.7	1.6

Tasks on Graphs

Graph-wise

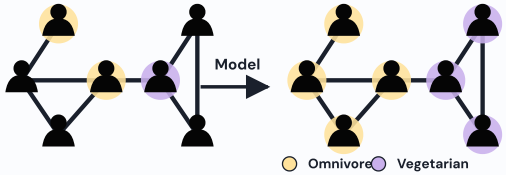
Predict a label (or scalar) for the whole graph.



Example: molecular property prediction, fraud detection, document classification.

Node-wise

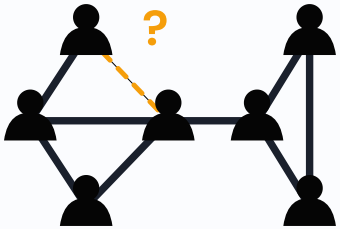
Predict labels (or scalars) for individual nodes.



Example: user attributes, protein function, paper topic prediction.

Edge-wise

Predict labels (or scalars) for edges. Link prediction is the special case of scoring missing or future edges.



Example: recommendations, knowledge graph completion, social tie prediction.

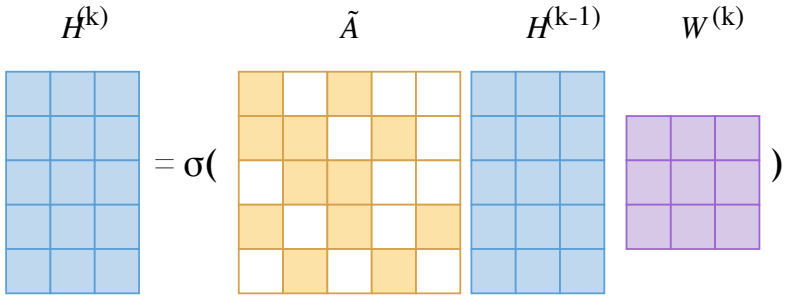
SECTION 03

(Static) Graph Neural Networks

Graph Neural Networks (GNNs)

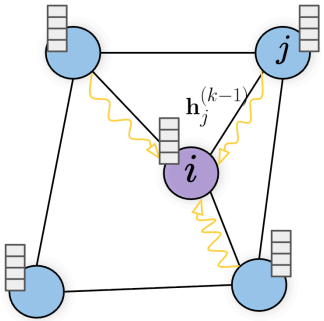
Convolutional GNN [7]

$$\mathbf{H}^{(k)} = \sigma(\tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)})$$
$$\mathbf{H}^{(0)} = \mathbf{X}$$



Message-Passing GNN [8]

$$\mathbf{m}_i^{(k)} = \mathrm{AGG}^{(k)}\left(\{\{\mathbf{h}_j^{(k-1)} : (i,j) \in E\}\}\right)$$
$$\mathbf{h}_i^{(k)} = \mathrm{COM}^{(k)}\left(\mathbf{h}_i^{(k-1)}, \mathbf{m}_i^{(k)}\right)$$
$$\mathbf{h}_i^{(0)} = \mathbf{x}_i$$



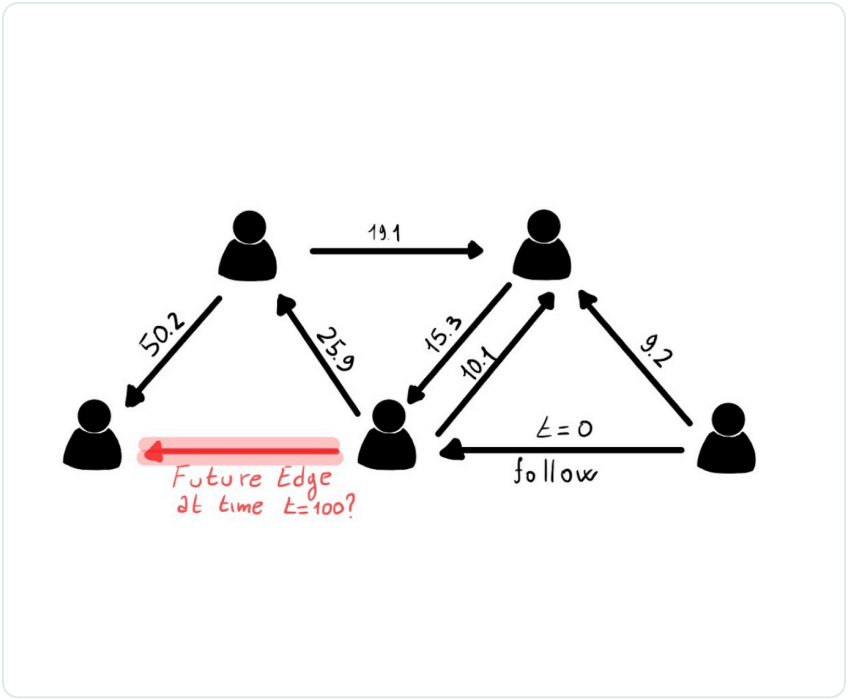
[7] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", ICLR 2017
[8] J. Gilmer et al., "Neural Message Passing for Quantum Chemistry", ICML 2017

SECTION 04

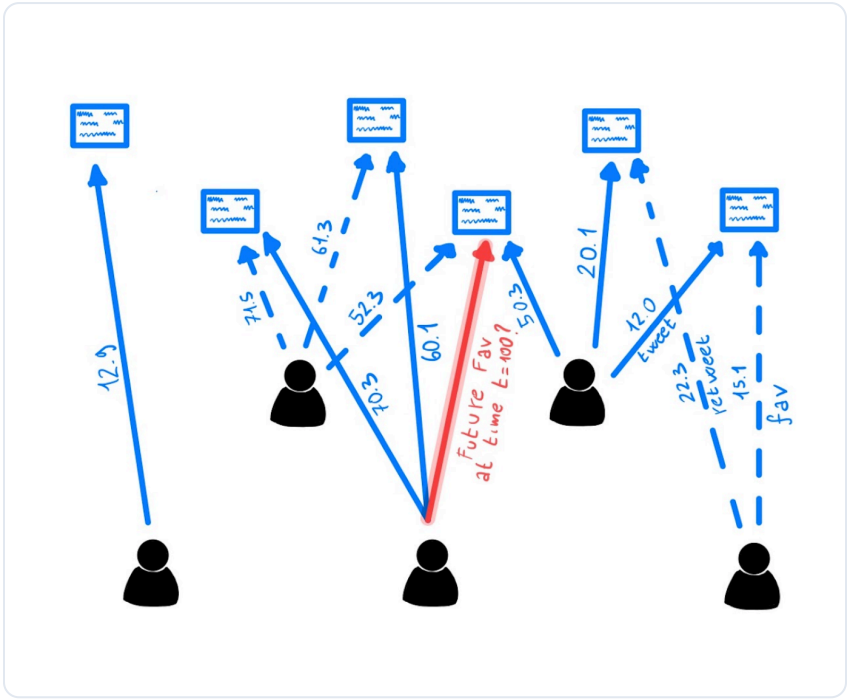
Dynamic Graphs

Some Examples of (Continuous) Dynamic Graphs

Graphs changing over time



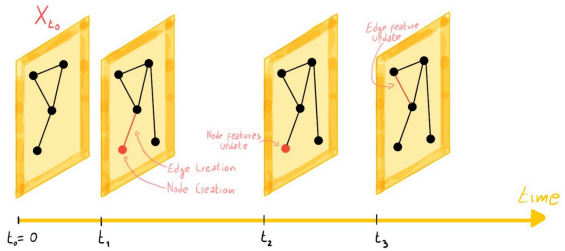
Social Networks

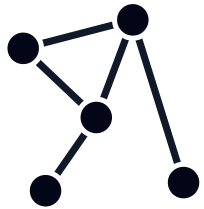


Interaction Networks

Why static GNNs are not enough

Collapsing event history into one snapshot loses order, timing and evolution





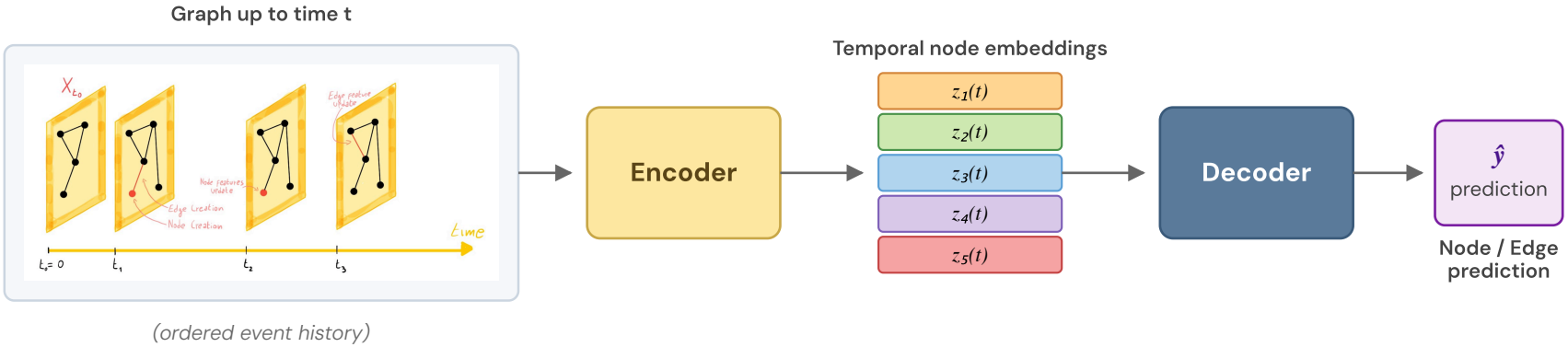
latest snapshot only

- *Information loss*: the last snapshot hides the evolution path, ie. *what* happened and *when*
- *Inefficiency*: every new event forces repeated computation
- *No timing prediction*: static GNNs do not support predicting when something will happen

SECTION 05

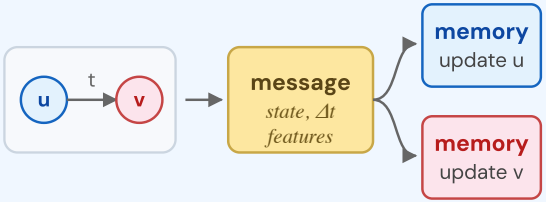
Models

Temporal Graph Model



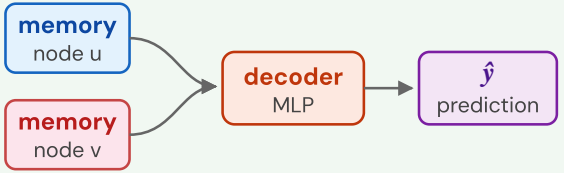
Memory-based Models [11, 12]

UPDATE TIME



Edge (u,v) builds a message from partner state, features & Δt; only the two touched nodes update their memory.

PREDICT TIME



Memories serve directly as node embeddings, fed into decoder with no extra computation at query time.

[11] R. Trivedi et al., "DyRep: Learning Representations over Dynamic Graphs", ICLR 2019
[12] S. Kumar et al., "Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks", KDD 2019

Memory-based Models: Pros and Cons

A strong online model, but not yet a full temporal graph encoder

PROS

- Strong **sequentiality inductive bias**
- Cheap **online updates** after each new event

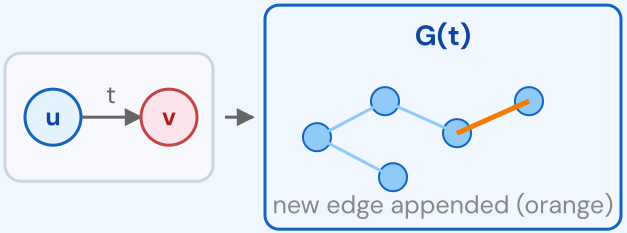
CONS

- Inactive nodes can become *stale*
- Graph context is mostly **local or indirect**
- Forced to process previous edges in sequential order

Graph-based Models [13]

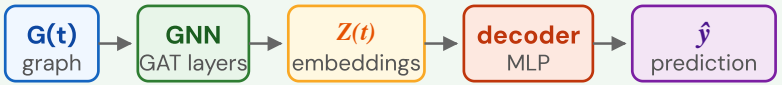
GNN on the graph of previous interactions, with timestamps as edge features

UPDATE TIME



Edge (u,v,t) is simply appended to G(t) — no computation performed at event time.

PREDICT TIME



GNN runs on full G(t) to produce node embeddings; must re-run for every new query — expensive at inference.

[13] D. Xu et al., "Inductive Representation Learning on Temporal Graphs", ICLR 2020

Graph-based Models: Pros and Cons [13]

Explicit graph structure mitigates staleness, but the GNN must re-run on every query

PROS

- No need for sequential processing in training
- Using the **graph explicitly** → mitigates staleness problem

CONS

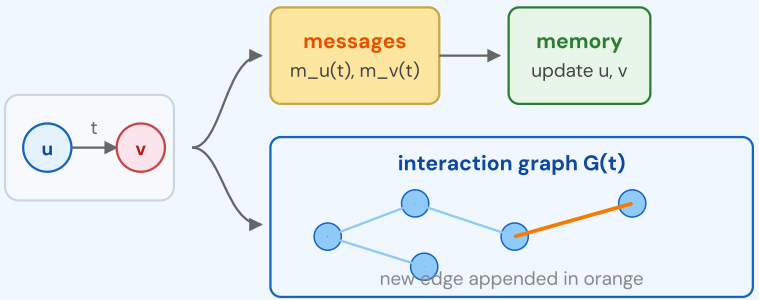
- Can only handle *edge addition* events
- Need to re-run GNN after each new event → **inefficient at inference**

[13] D. Xu et al., "Inductive Representation Learning on Temporal Graphs", ICLR 2020

TGN: Temporal Graph Networks [14]

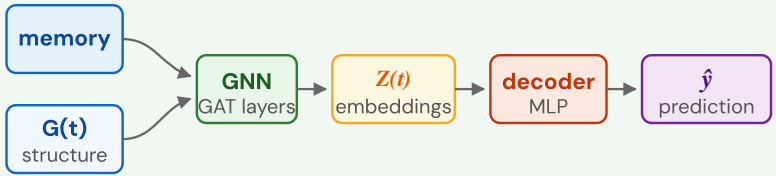
Our work: a modular framework that combines memory-based and graph-based temporal learning

UPDATE TIME



The event is its own object: it creates messages, updates memories for u and v, and is also appended to G(t).

PREDICT TIME



Memory and graph structure are inputs to the GNN, which produces Z(t) for the decoder.

[14] **E. Rossi** et al., "Temporal Graph Networks for Deep Learning on Dynamic Graphs", ICML Workshop on Graph Representation Learning and Beyond 2020

TGN: Best of Both Worlds

Memory-based online updates plus graph-based structural context

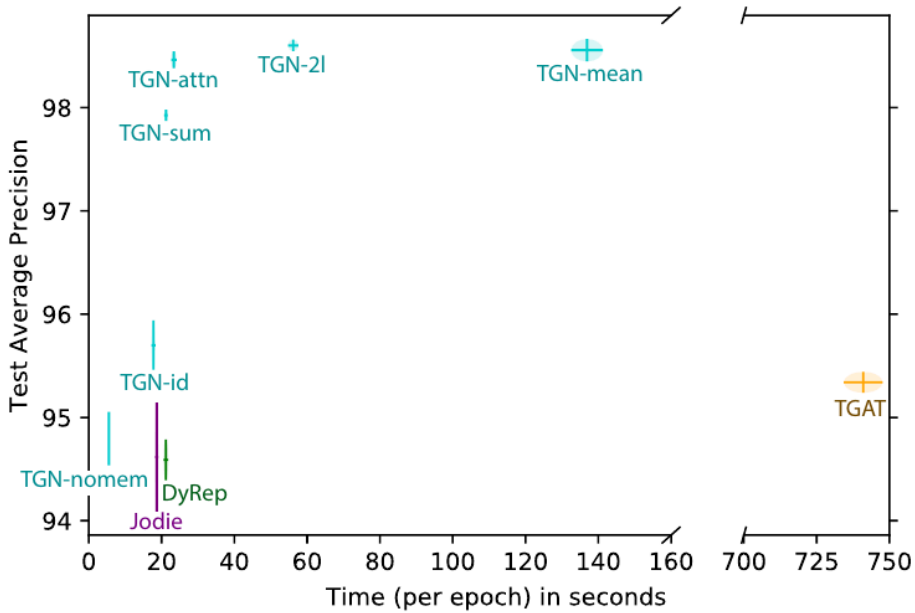
PROS

- **Prediction-time GNN** can be cheaper (shallower) thanks to leveraging the **memory**
- **Graph context** at prediction time mitigates stale local state
- Combines **sequentiality inductive bias** with **topological information**
- One **modular notation** covers memory-based and graph-based models

[14] **E. Rossi** et al., "Temporal Graph Networks for Deep Learning on Dynamic Graphs", ICML Workshop on Graph Representation Learning and Beyond 2020

TGN: State-of-the-Art in 2020

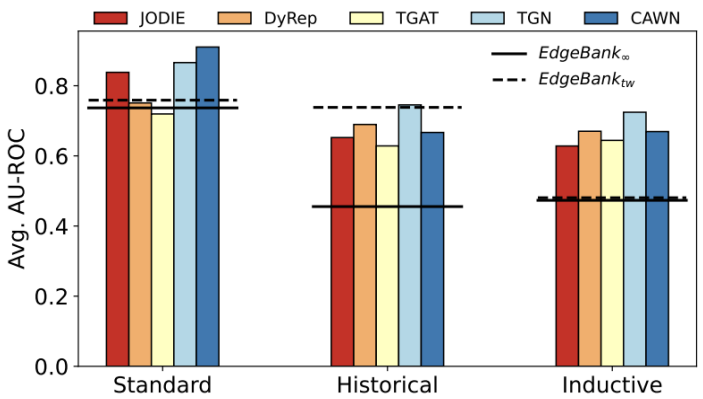
High accuracy with much lower per-epoch cost than the previous methods



[14] **E. Rossi** et al., "Temporal Graph Networks for Deep Learning on Dynamic Graphs", ICML Workshop on Graph Representation Learning and Beyond 2020

TGN: Still a Strong Baseline Today

Years of follow-up work, yet TGN remains hard to beat on standard benchmarks [15, 16]



Datasets	Flickr	YouTube	Patent	WikiLink
JODIE	46.21 ± 0.83	41.67 ± 2.86	24.60 ± 0.38	57.94 ± 1.33
DyRep	38.04 ± 4.19	35.12 ± 4.13	21.01 ± 1.14	42.63 ± 1.33
TGAT	23.53 ± 3.35	43.56 ± 2.53	8.49 ± 0.18	OOT
TGN	46.03 ± 6.78	55.16 ± 5.89	22.83 ± 2.25	62.94 ± 2.16
CAWN	48.69 ± 6.08	47.55 ± 1.08	12.34 ± 0.47	OOT
TCL	40.00 ± 1.76	50.17 ± 1.98	10.60 ± 1.75	43.02 ± 2.16
GraphMixer	45.01 ± 0.08	58.87 ± 0.12	18.97 ± 2.54	48.57 ± 0.02
DyGFormer	49.58 ± 2.87	46.08 ± 3.44	14.20 ± 2.93	OOT

[15] F. Poursafaei et al., "Towards Better Evaluation for Dynamic Link Prediction", NeurIPS Datasets and Benchmarks Track 2022
[16] L. Yi et al., "TGB-Seq Benchmark: Challenging Temporal GNNs with Complex Sequential Dynamics", ICLR 2025

SECTION 06

Graph Machine Learning on the Bitcoin Lightning Network

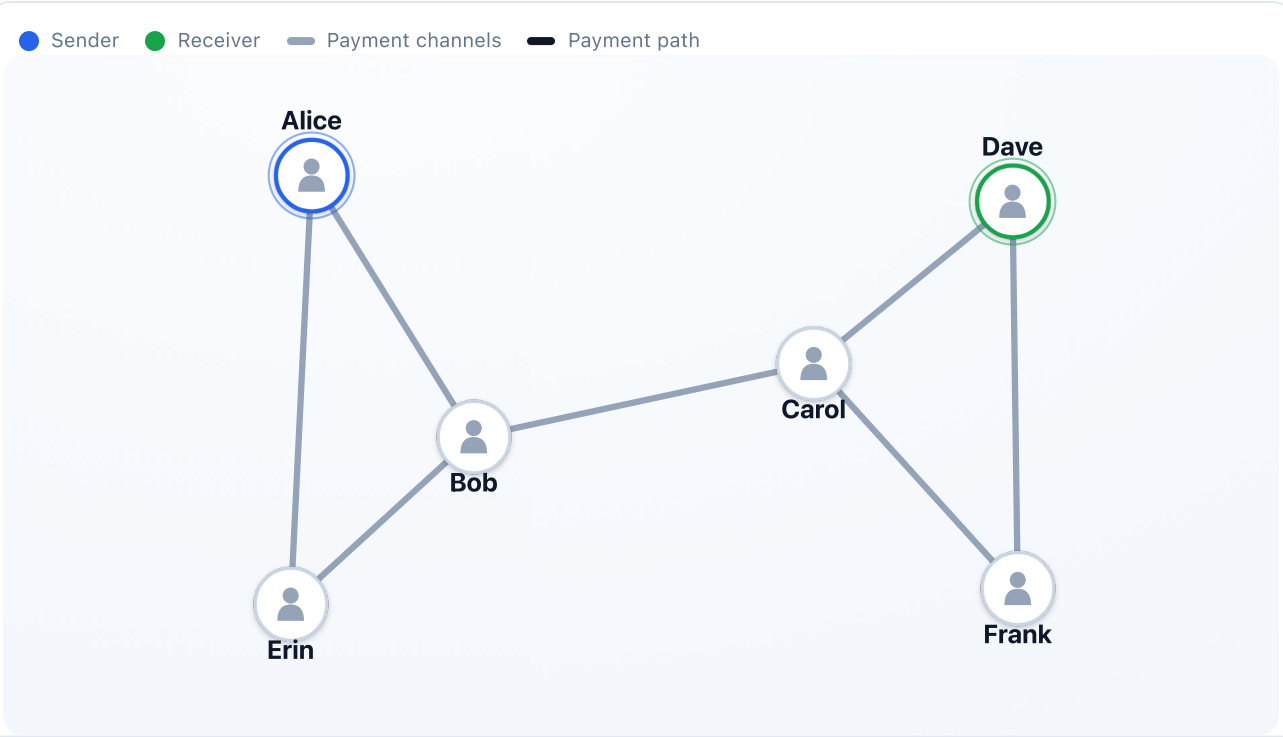
Lightning Network: Motivation

Making Bitcoin scalable for everyday payments

- **Problem:** BTC has a bottleneck of ~5 transactions/sec, 1000x less than Mastercard.
- **Solution:** The Lightning Network enables 5M+ transactions/sec through fast, low-cost off-chain payments.
- **Why it matters:** Makes BTC more practical for everyday payments, micropayments, and future agent-to-agent transactions.
- **Adoption:** ~$15B in 2025, with 300% YoY growth.

Lightning Network Basics

Payments move through channels, one hop at a time



From

Alice

To

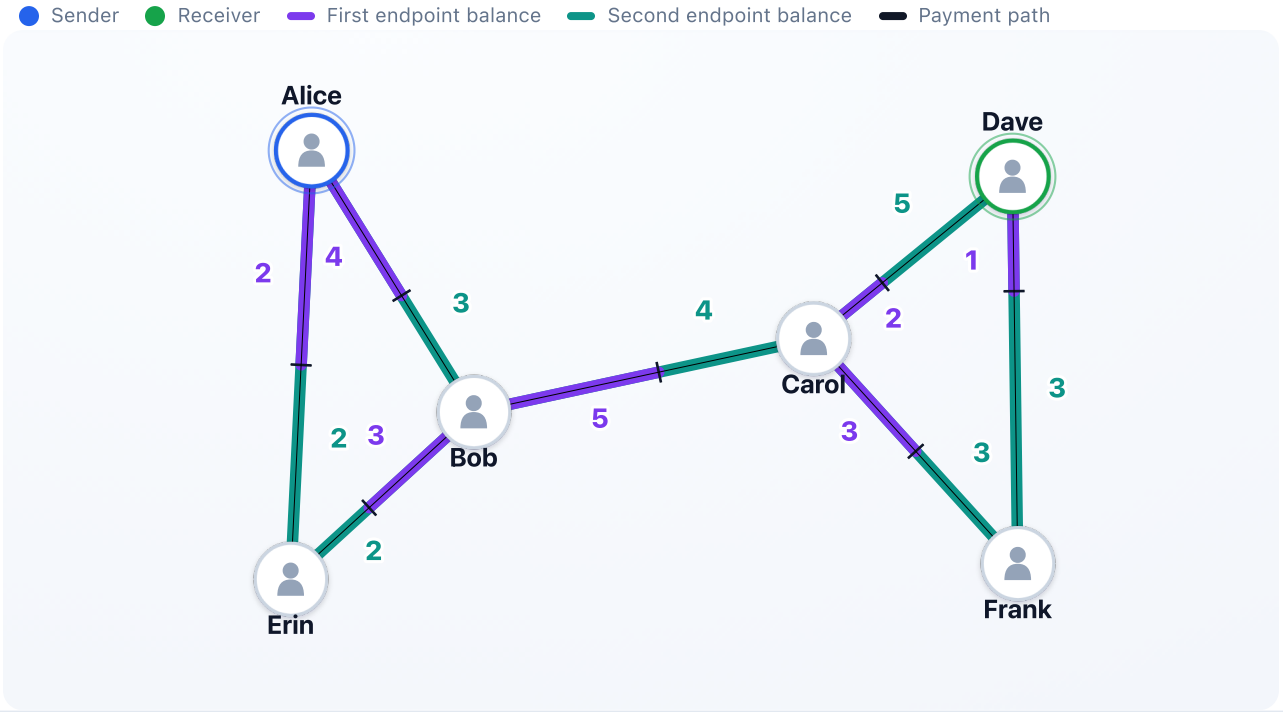
Dave

Amount — 2 ฿

Send payment

Lightning Network: Balances and Payments

Payments move only where liquidity exists in the right direction



From

Alice

To

Dave

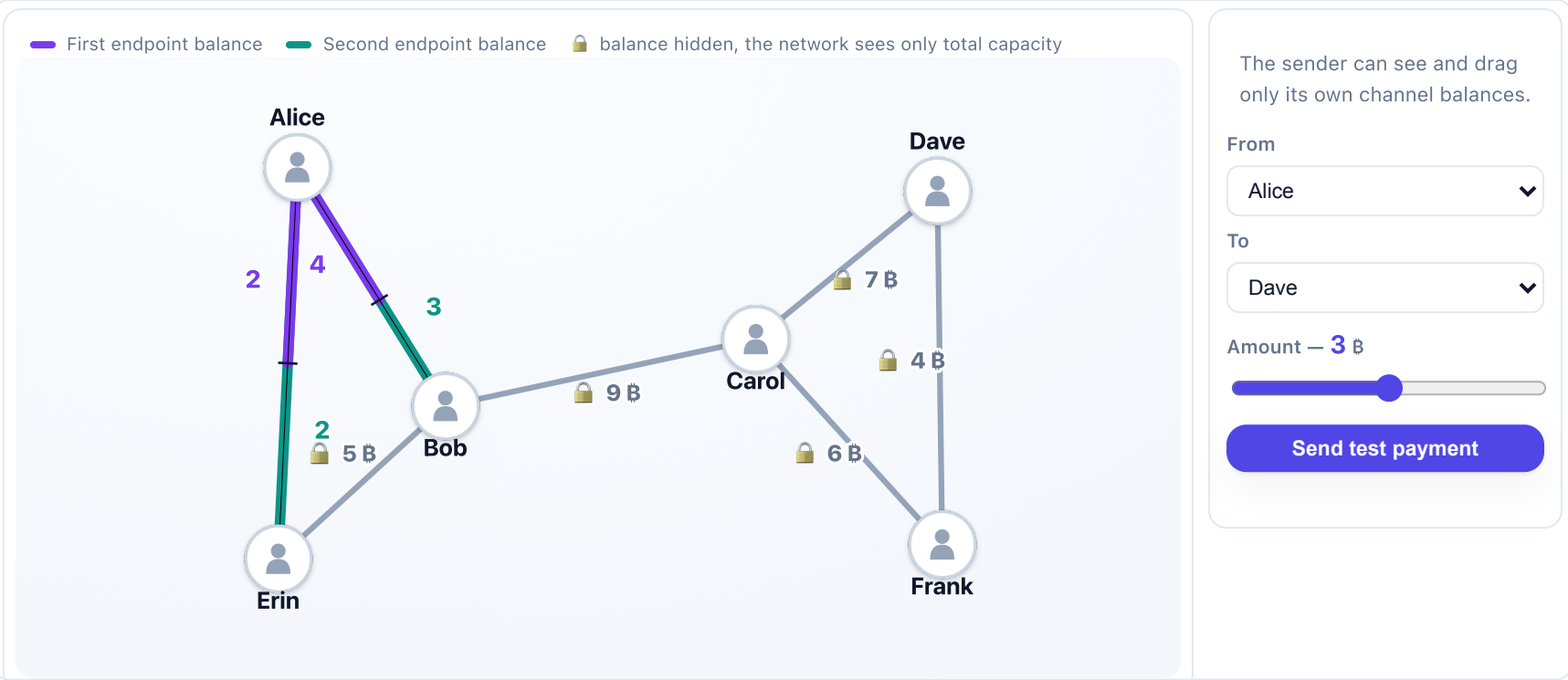
Amount — 2 ฿

Send payment

Drag any channel split to change the starting liquidity.

Problem 1: Channel Balances are Actually Hidden

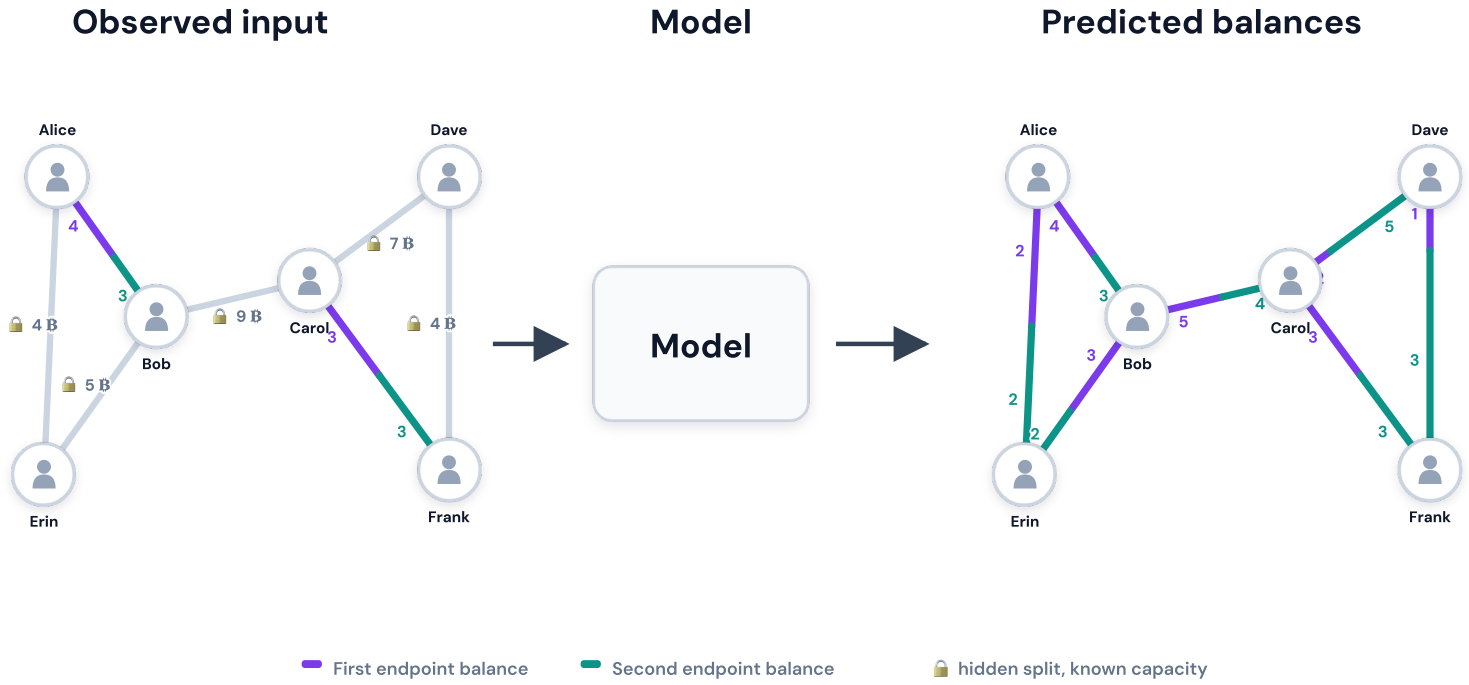
A node can only see the directional balances of its channels



[19] V. Davis, ..., **E. Rossi**, "Channel Balance Interpolation in the Lightning Network via Machine Learning", IEEE International Conference on Blockchain and Cryptocurrency (ICBC), arXiv 2405.12087 2025

Channel Balance Interpolation

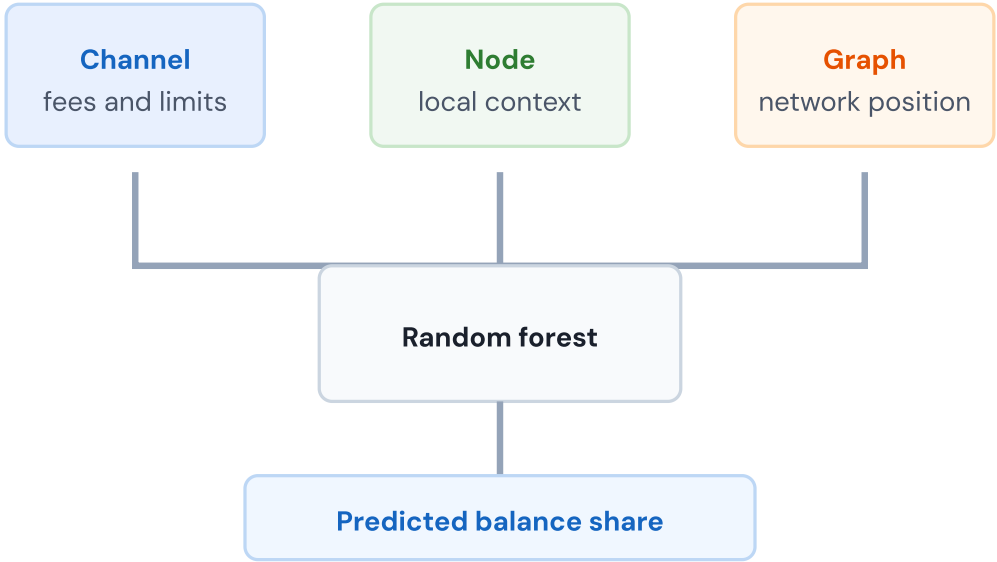
RQ: Can we predict the hidden directional balances for the network channels?



[19] V. Davis, ..., **E. Rossi**, "Channel Balance Interpolation in the Lightning Network via Machine Learning", IEEE International Conference on Blockchain and Cryptocurrency (ICBC), arXiv 2405.12087 2025

Balance Interpolation: Methods

Random forests over public node, channel and graph features

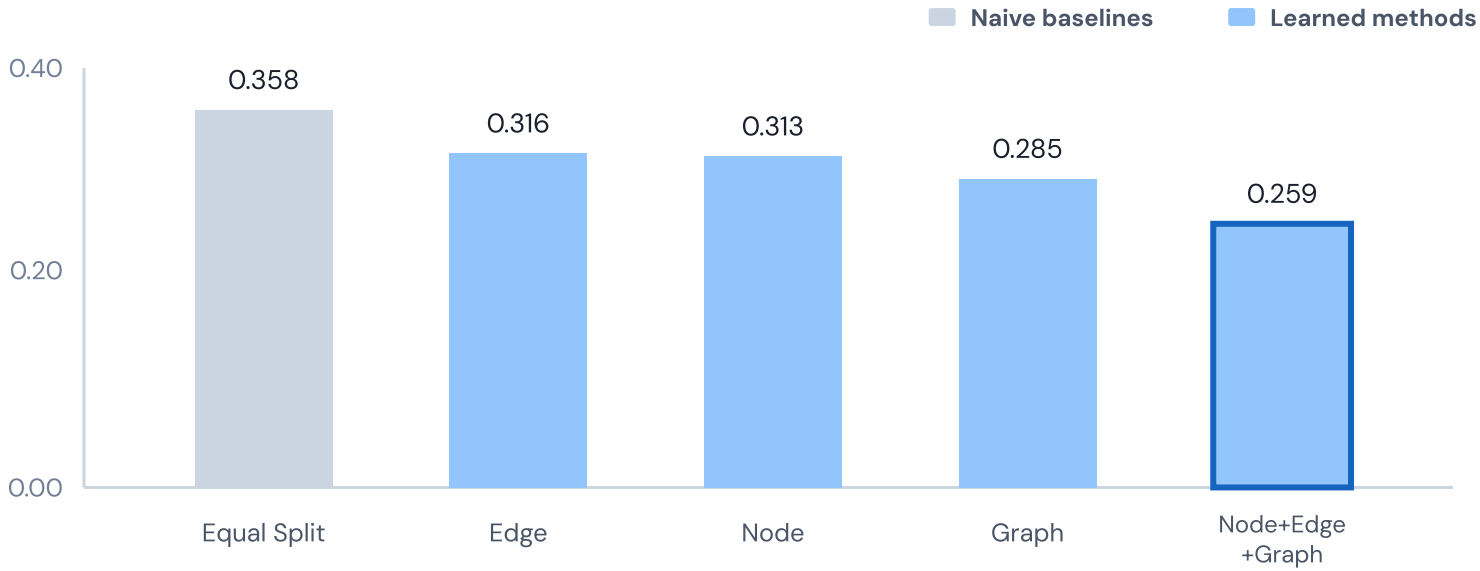


[19] V. Davis, ..., **E. Rossi**, "Channel Balance Interpolation in the Lightning Network via Machine Learning", IEEE International Conference on Blockchain and Cryptocurrency (ICBC), arXiv 2405.12087 2025

Balance Interpolation: Results

The model combining node, edge and graph features improves by 27% over simple heuristics

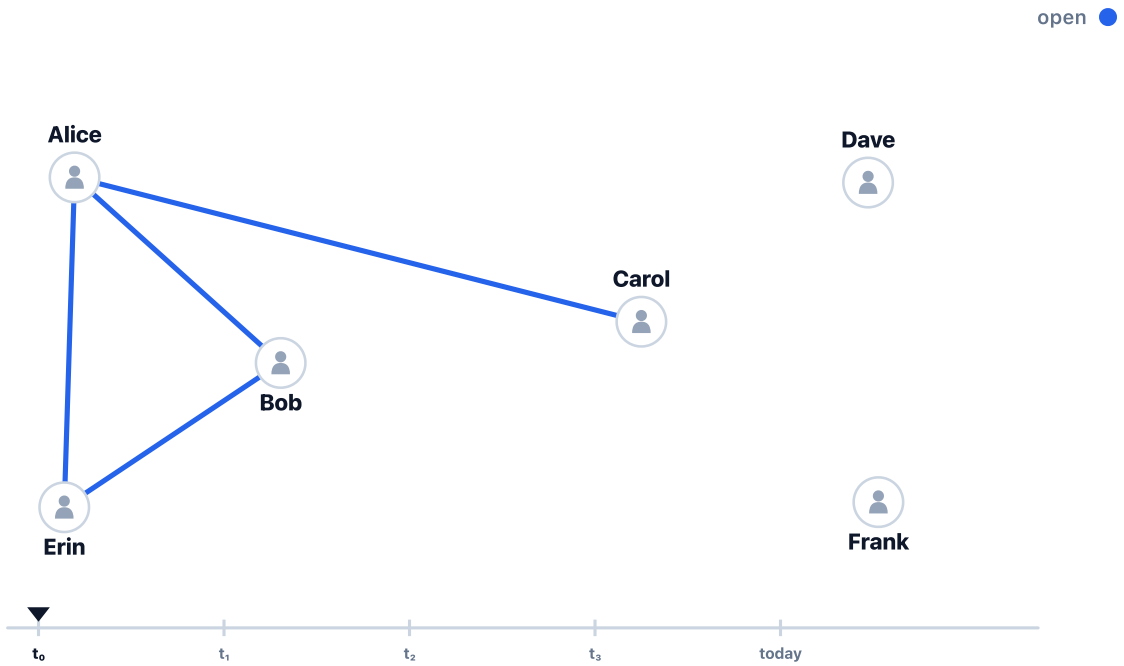
BALANCE-SHARE MAE, LOWER IS BETTER



[19] V. Davis, ..., **E. Rossi**, "Channel Balance Interpolation in the Lightning Network via Machine Learning", IEEE International Conference on Blockchain and Cryptocurrency (ICBC), arXiv 2405.12087 2025

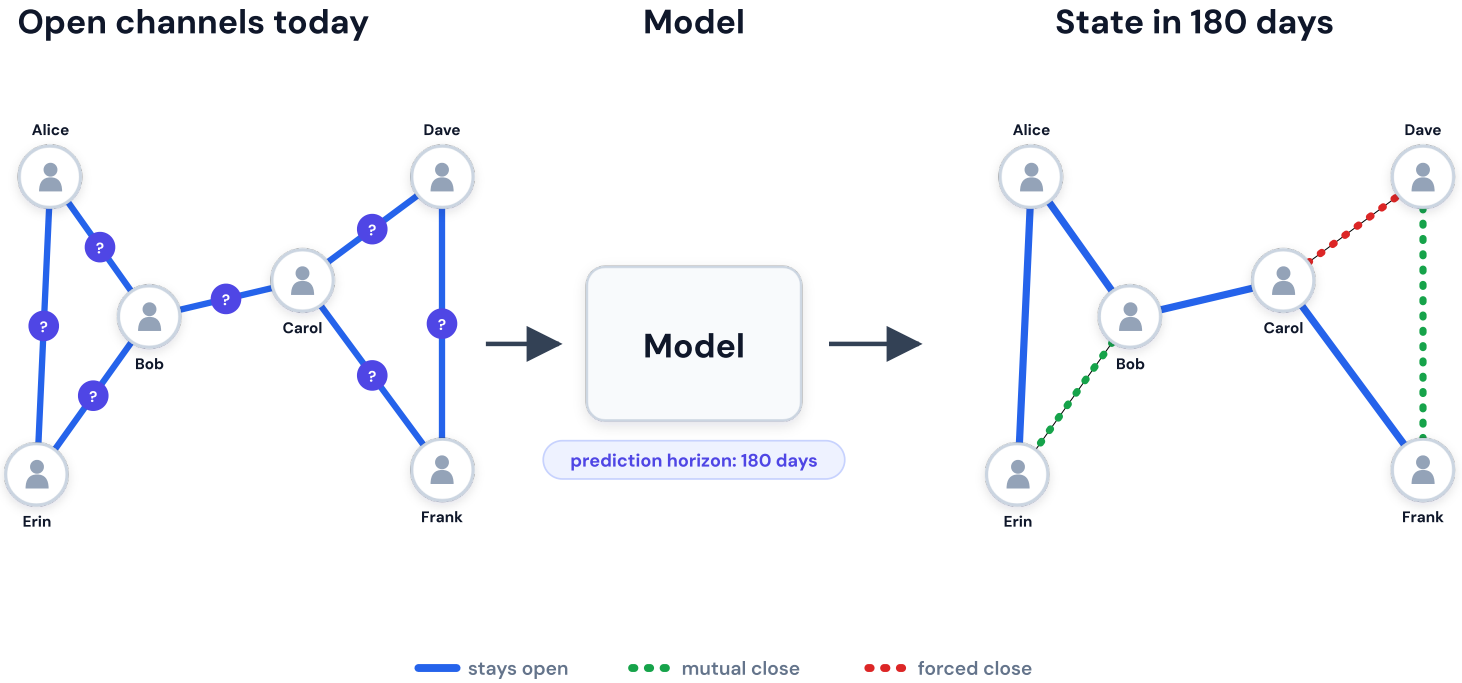
Problem 2: The Lightning Network Changes over Time

New edges can be opened, mutually closed, or forced closed



Channel Closure Prediction

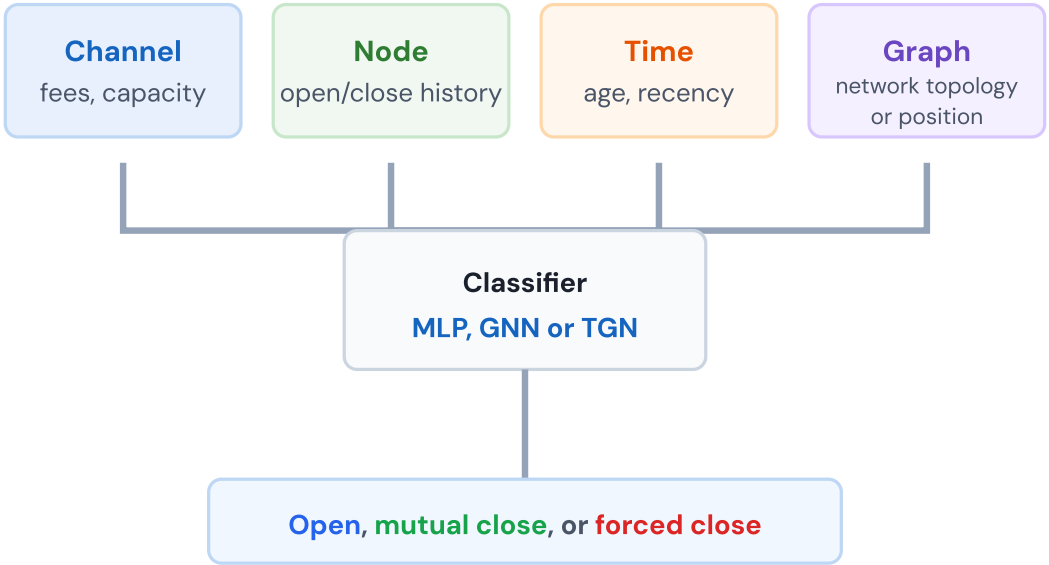
RQ: Which channels will stay open, close mutually, or force close?



[20] S. Antonelli, ..., **E. Rossi**, "Predicting Channel Closures in the Lightning Network with Machine Learning", International Conference on Blockchain Computing and Applications (BCCA), arXiv 2605.12759 2026

Closure Prediction: Methods

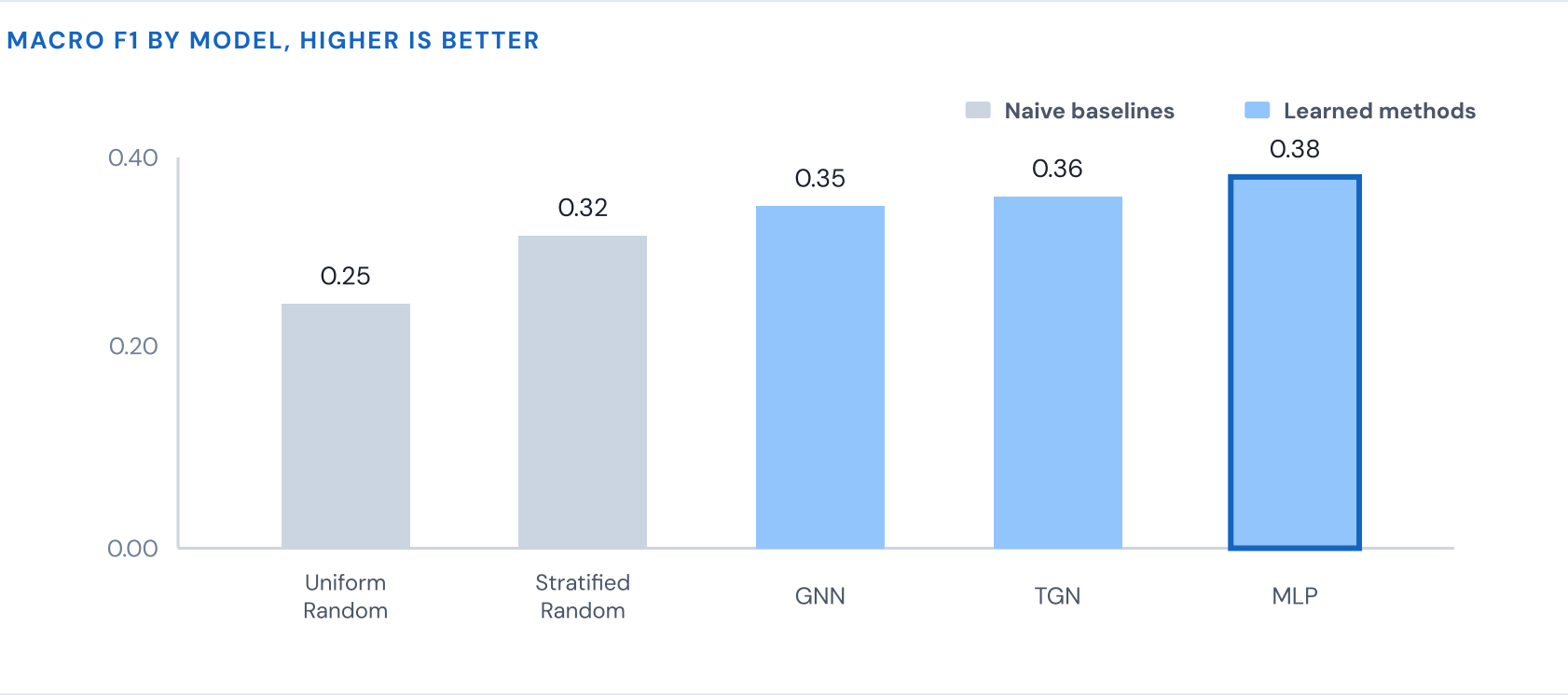
Temporal link classification given past history



[20] S. Antonelli, ..., **E. Rossi**, "Predicting Channel Closures in the Lightning Network with Machine Learning", International Conference on Blockchain Computing and Applications (BCCA), arXiv 2605.12759 2026

Closure Prediction: Results

A simple MLP beats graph-aware models, but the overall performance is modest

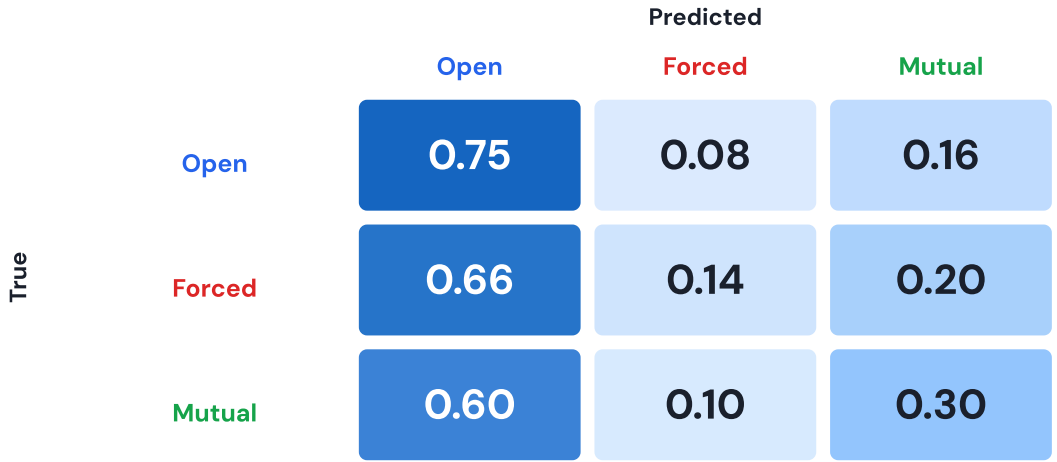


[20] S. Antonelli, ..., **E. Rossi**, "Predicting Channel Closures in the Lightning Network with Machine Learning", International Conference on Blockchain Computing and Applications (BCCA), arXiv 2605.12759 2026

The Model Struggles to Predict Closures

Public information lacks balances, payment failures and uptime, the private signals that would be most useful here

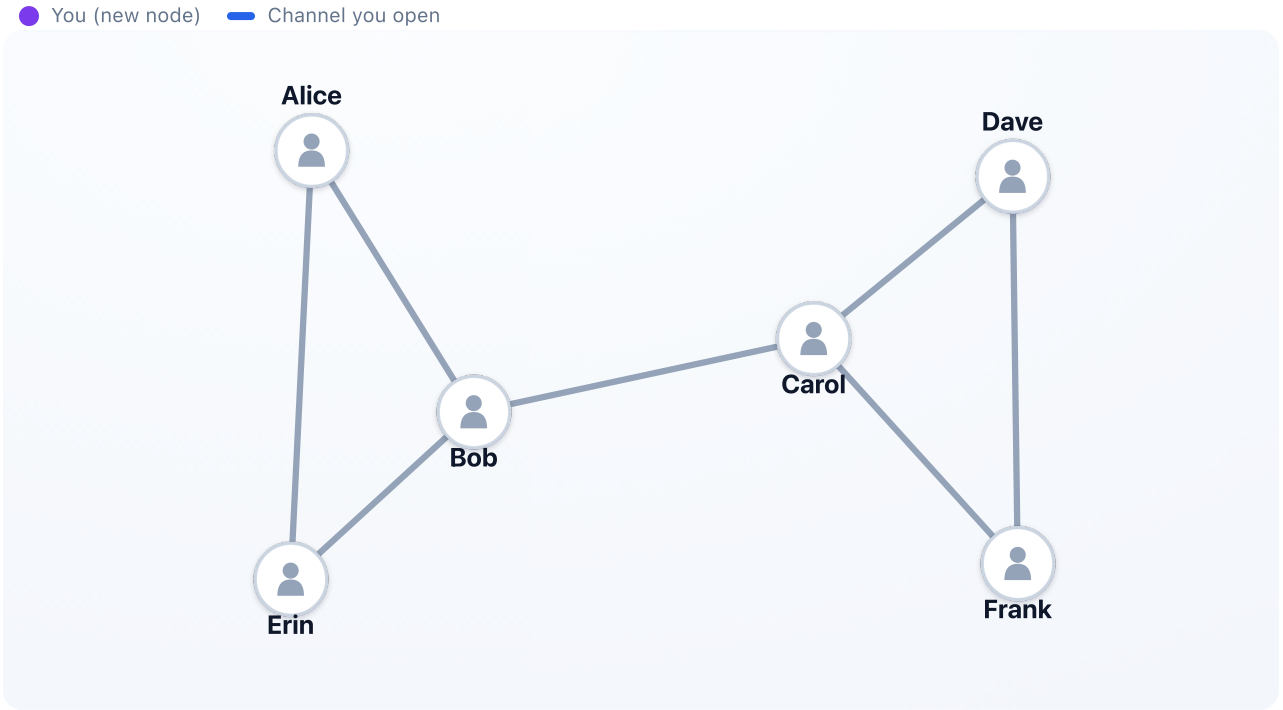
NORMALIZED CONFUSION MATRIX FOR THE MLP



[20] S. Antonelli, ..., **E. Rossi**, "Predicting Channel Closures in the Lightning Network with Machine Learning", International Conference on Blockchain Computing and Applications (BCCA), arXiv 2605.12759 2026

Problem 3: How to optimally allocate a given budget?

When joining the network, a node has to choose who to connect to and how to allocate its budget



One big channel

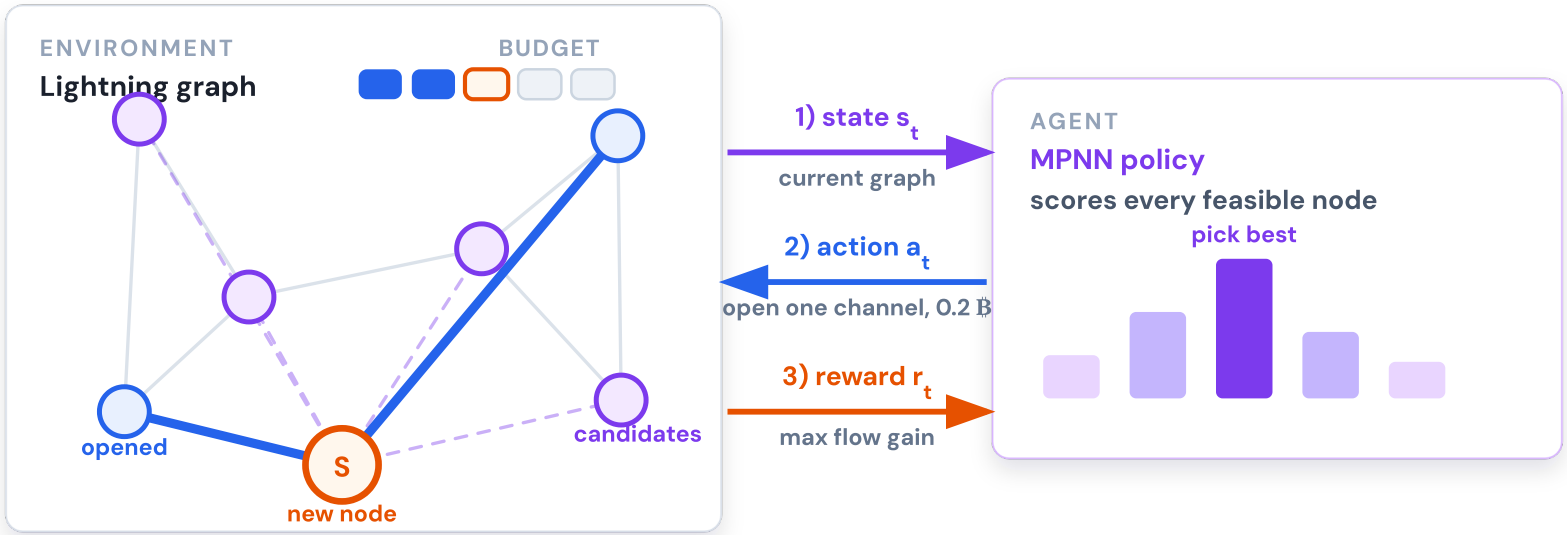
Link left to right

Connect to everyone

[21] H. Rush, ..., **E. Rossi**, "MP-Flow: Learning Budgeted Max-Flow Optimization on the Lightning Network with Deep Graph Reinforcement Learning", Manuscript 2026

MP-Flow: Methods

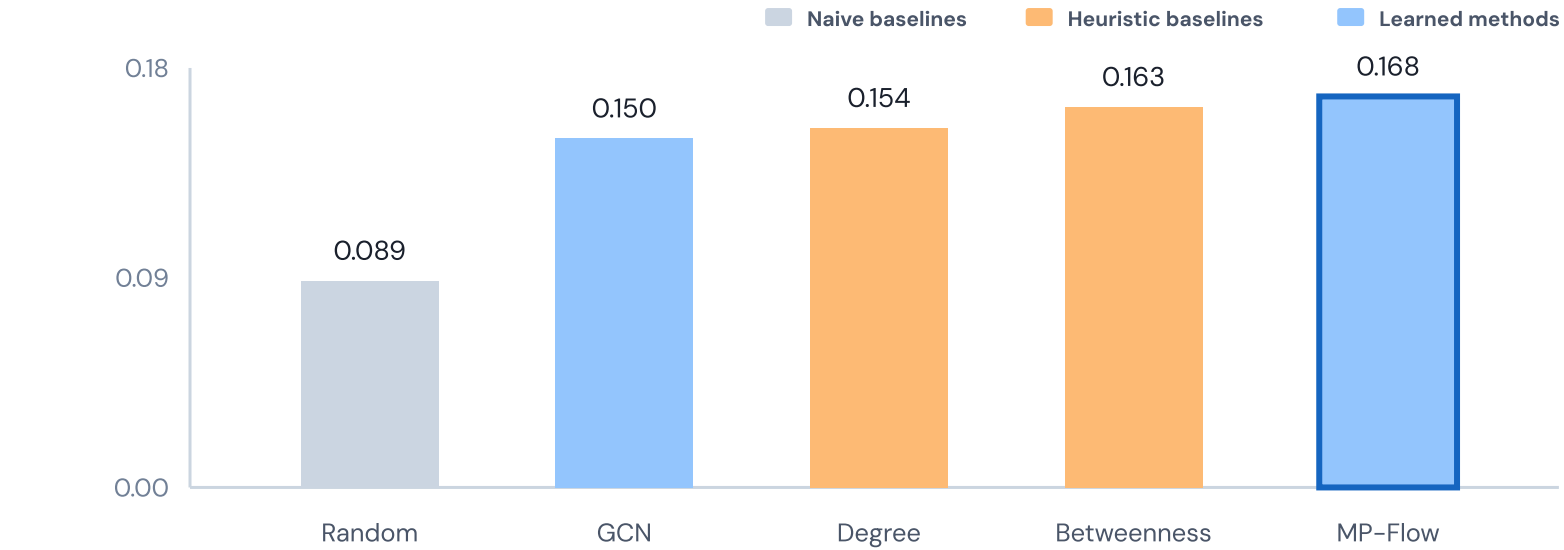
Channel placement as reinforcement learning: a graph policy picks peers one at a time



[21] H. Rush, ..., **E. Rossi**, "MP-Flow: Learning Budgeted Max-Flow Optimization on the Lightning Network with Deep Graph Reinforcement Learning", Manuscript 2026

MP-Flow: Results

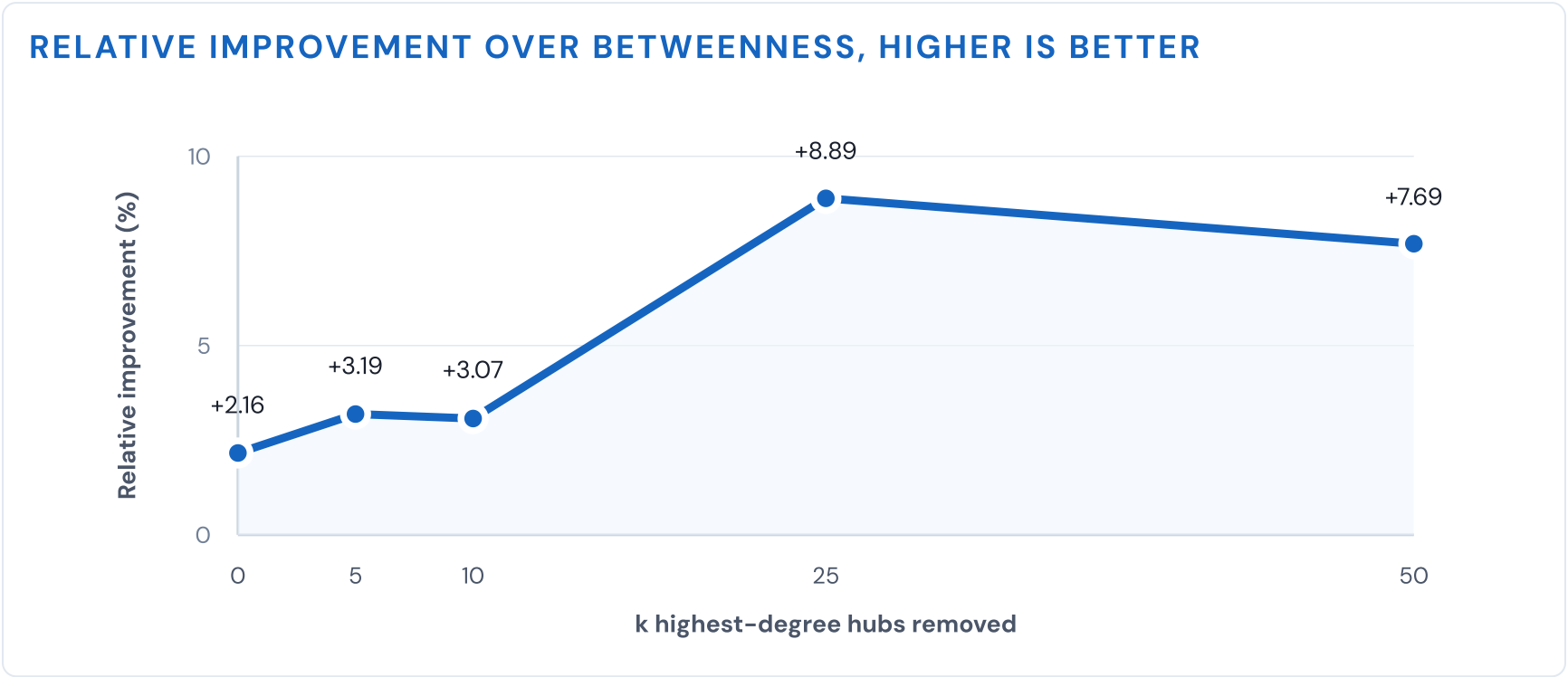
MP-Flow outperforms all other methods, with a 88.5% relative gain over a random baseline



[21] H. Rush, ..., **E. Rossi**, "MP-Flow: Learning Budgeted Max-Flow Optimization on the Lightning Network with Deep Graph Reinforcement Learning", Manuscript 2026

MP-Flow is learning more than just connecting to big hubs

Relative improvement over Betweenness grows as top-degree hubs are pruned



[21] H. Rush, ..., **E. Rossi**, "MP-Flow: Learning Budgeted Max-Flow Optimization on the Lightning Network with Deep Graph Reinforcement Learning", Manuscript 2026

MP-Flow is in production at Amboss

The learned policy now powers peer recommendations for real Lightning channel openings

DECISIONS EXECUTED

4,640

channel-open actions selected by the agent

CAPITAL ALLOCATED

267.3 ฿

over $16M deployed into Lightning channels

MANAGED NODES

30

managed production nodes receiving recommendations

[21] H. Rush, ..., **E. Rossi**, "MP-Flow: Learning Budgeted Max-Flow Optimization on the Lightning Network with Deep Graph Reinforcement Learning", Manuscript 2026

Conclusion: main points

Graphs are everywhere, and most real financial graphs are dynamic

DYNAMIC GRAPHS

Graphs are everywhere, and most graphs, particularly in finance, are dynamic.

TEMPORAL GRAPH MODELS

Temporal Graph Models leverage the sequence and timing of events, on top of the topology.

LIGHTNING NETWORK

An exciting application where we have worked on:

- Predicting the directional balance of channels
- Predicting whether channels will stay open or close
- Optimally selecting the channels to which to allocate liquidity

THANK YOU

Questions?

Graph Machine Learning for Dynamic Financial Networks · EPFL Workshop on Graph Learning in Financial Networks · 10/07/2026