IEEE CROATIA - INVITED TALK – ZAGREB, 08 NOV 2023

Causal Temporal GNNs as Decentralized Memory Networks

Lodovico Giaretta lodovico.giaretta@ri.se

RI. SE

About Me

Researcher @ Computer Science department, RISE Research Institutes of Sweden

Ph.D. in ICT @ KTH Royal Institute of Technology, Stockholm

- graduated: June 2023
- thesis: “Towards Decentralized Graph Learning”

Research Interests:

- Decentralized machine learning
- Graph representation learning (GRL)
- Adaptive, scalable, privacy-preserving and energy-efficient fully-decentralized GRL

About RISE

- Sweden’s state-owned research institute

- ~3300 employees: 4th largest research institute in Europe!

- Deep expertise: from academia to industry

- Wide expertise: from chemistry, to ship design, to computer science

Computer Science @ RISE

Cybersecurity

- Cyber Range testbed
- Vulnerability testing
- IoT security
- AI & Cyber
- Cyber Node

Internet of Things and 5G

- Battery-free IoT
- Secure IoT transfer
- 6G security

Datacenter

- Datacenter technologies
- Heat reuse & energy efficiency
- Cloud & Edge testbed

Data platforms

- AI & Earth observation data
- Digital twins
- Edge computing platforms
- High Performance Computing

AI and machine learning

- AI for network automation
- Resource- efficient ML
- Soundscape analysis
- Cross-lingual and Multilingual AI

Industrial data analysis

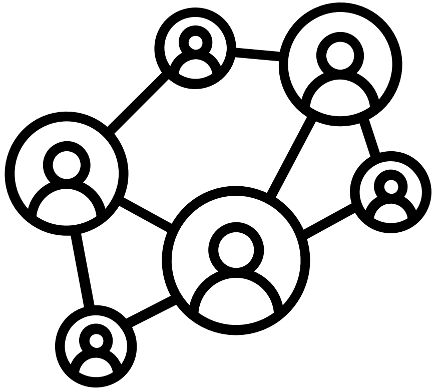
- Knowledge graphs and reasoning
- Predictive maintenance
- Causal inference
- Compilers

Agenda

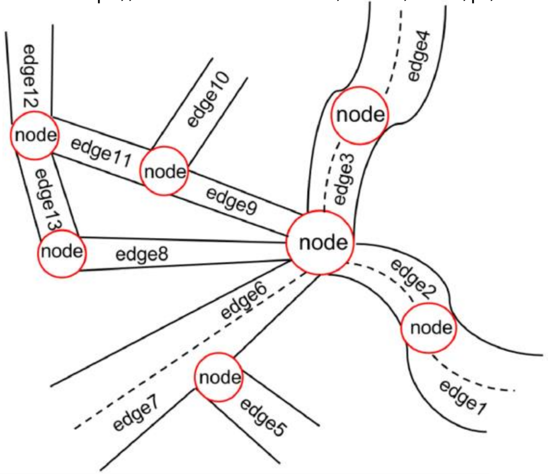
- Intro
  - Graph Representation Learning
  - (Causal) Temporal GNNs
  - Memory Networks
- Use Case (centralized)
- Towards Decentralization
- Conclusion

Intro

Graphs are Everywhere

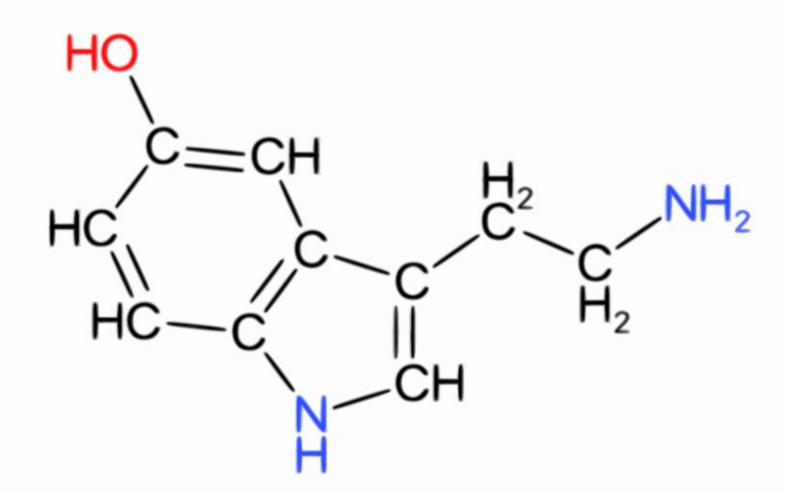


adapted from https://www.sciencedirect.com/science/article/pii/S0031320321003617



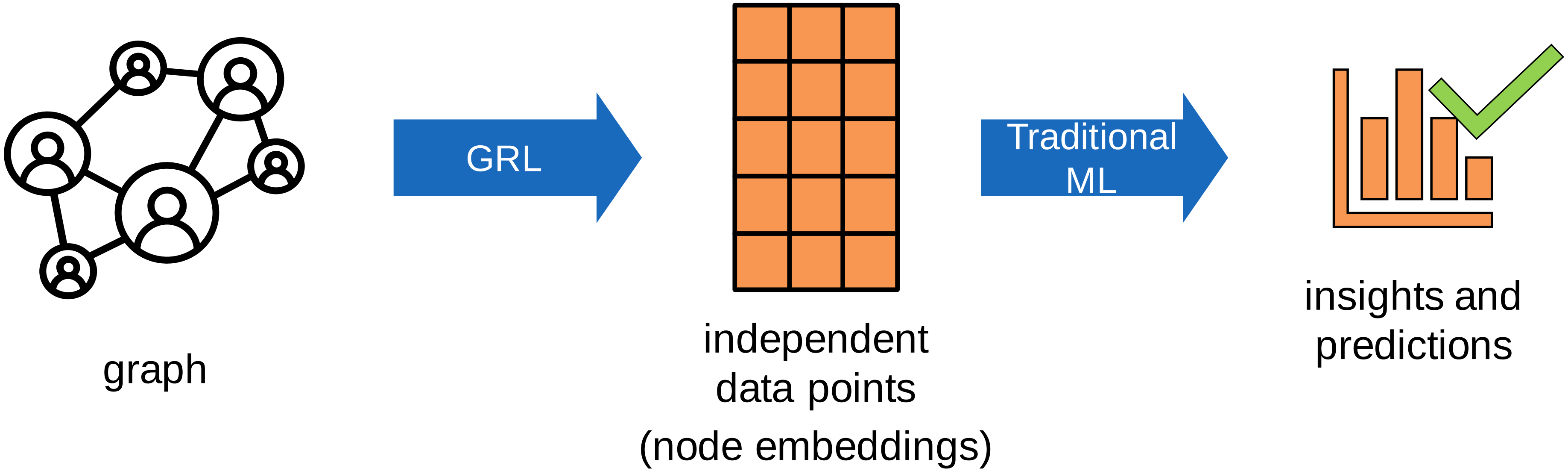
Isolated data points are a rarity!

As much information in the relations, if not more!



adapted from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008504

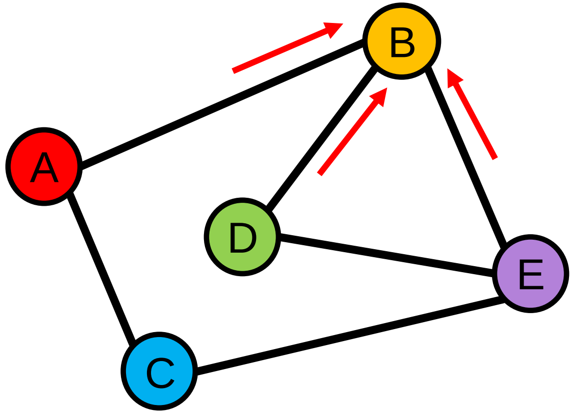
Graph Representation Learning



Graph Neural Networks

Core idea: embed each node based on the embeddings of its **neighbours**

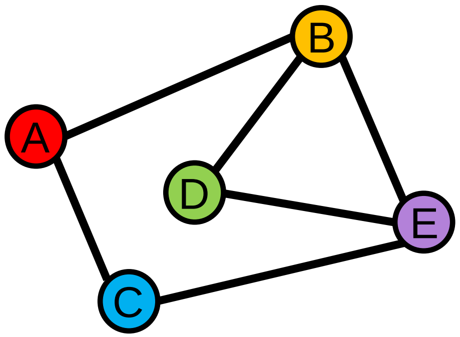
$$h_i = f\left(h_i, \left\{h_j \middle| \, j \in N(i)\right\}\right)$$

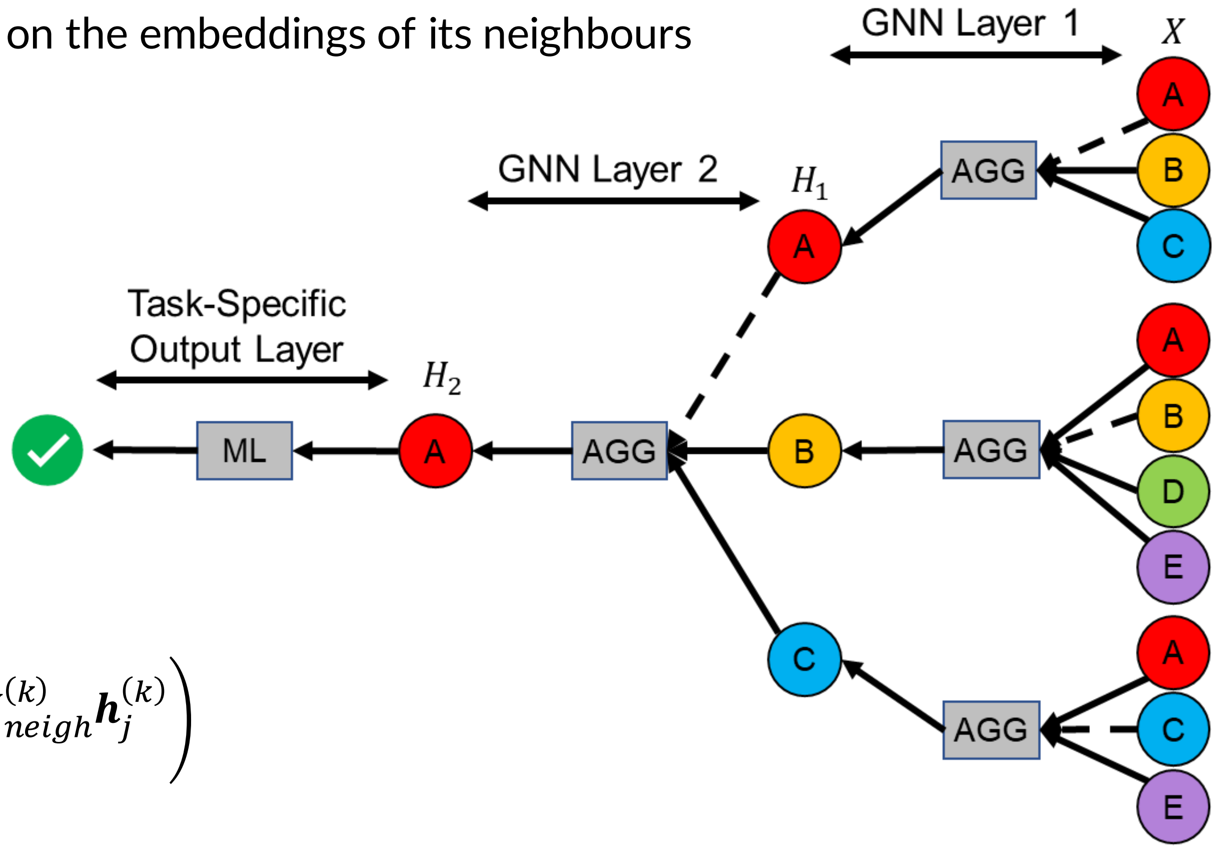


$$\boldsymbol{h}_i^{(k+1)} = \sigma\left(\boldsymbol{W}_{loc}^{(k)}\boldsymbol{h}_i^{(k)} + \sum_{j\in N(i)} \boldsymbol{W}_{neigh}^{(k)}\boldsymbol{h}_j^{(k)}\right)$$

Graph Neural Networks

Core idea: embed each node based on the embeddings of its neighbours

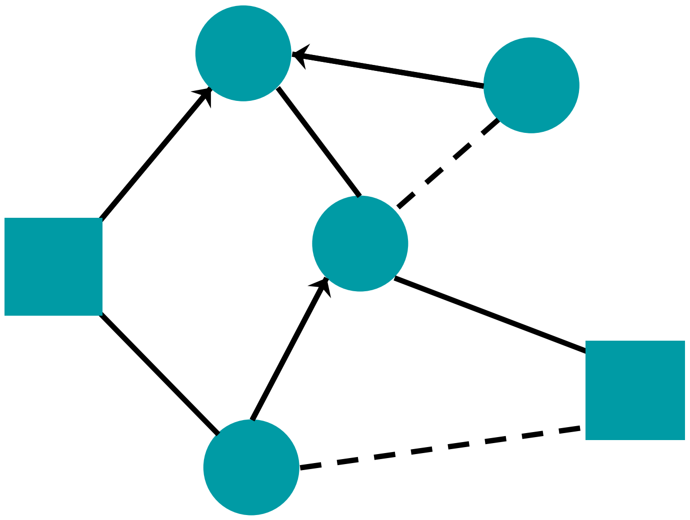




$$\boldsymbol{h}_i^{(k+1)} = \sigma\left(\boldsymbol{W}_{loc}^{(k)}\boldsymbol{h}_i^{(k)} + \sum_{j\in N(i)} \boldsymbol{W}_{neigh}^{(k)}\boldsymbol{h}_j^{(k)}\right)$$

A Zoo of Graphs (and GNNs)

- Homogeneous vs heterogeneous nodes/edges
- Directed vs undirected
- Bipartite
- Weighted
- Node vs edge features



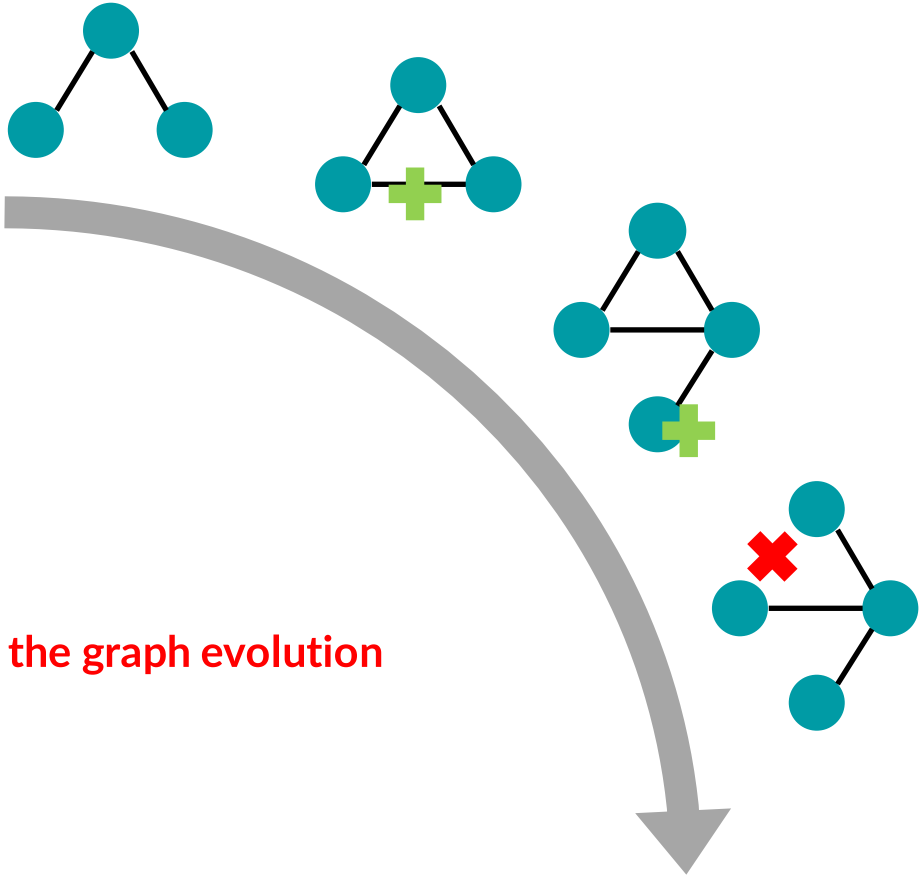
Most GNNs work on **static graphs**!

Dynamic Graphs

Few graphs are truly immutable!

Different kinds of changes:

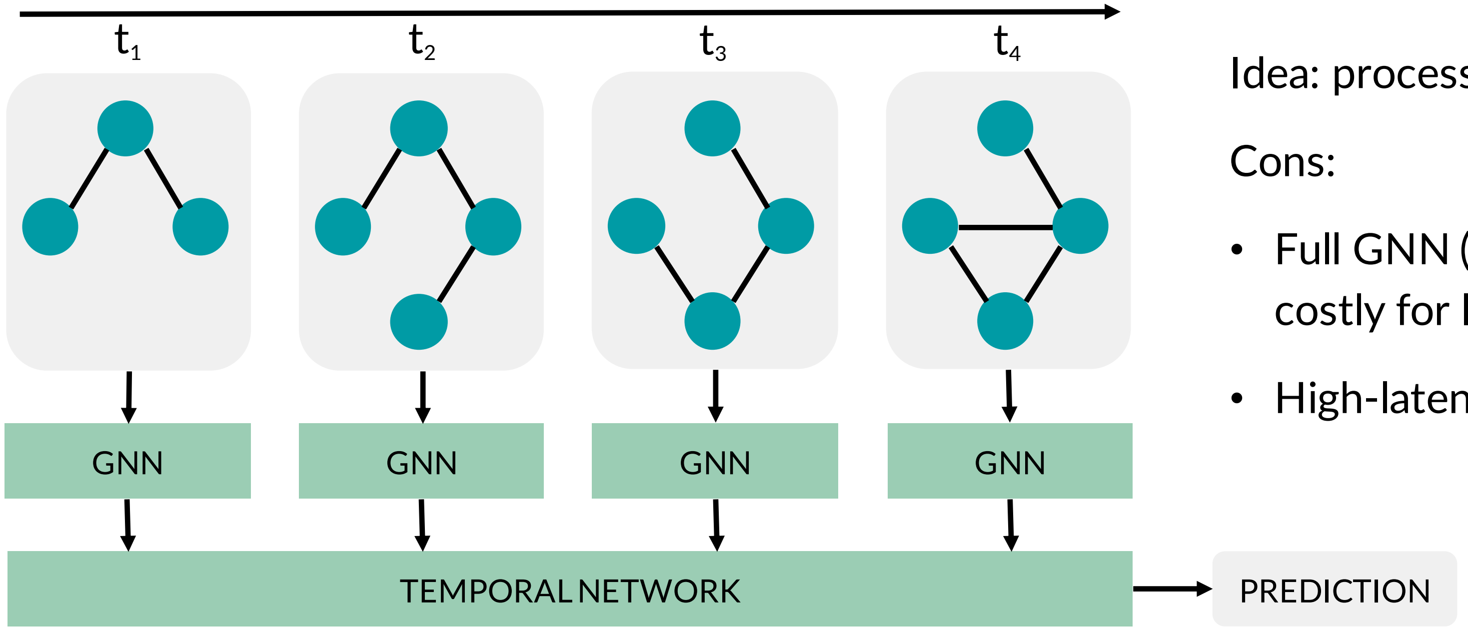
- Edge vs node changes
- Additions vs deletions



Often, it's useful to **understand, model and predict the graph evolution**

- Temporal GNNs!

Discrete Temporal GNNs

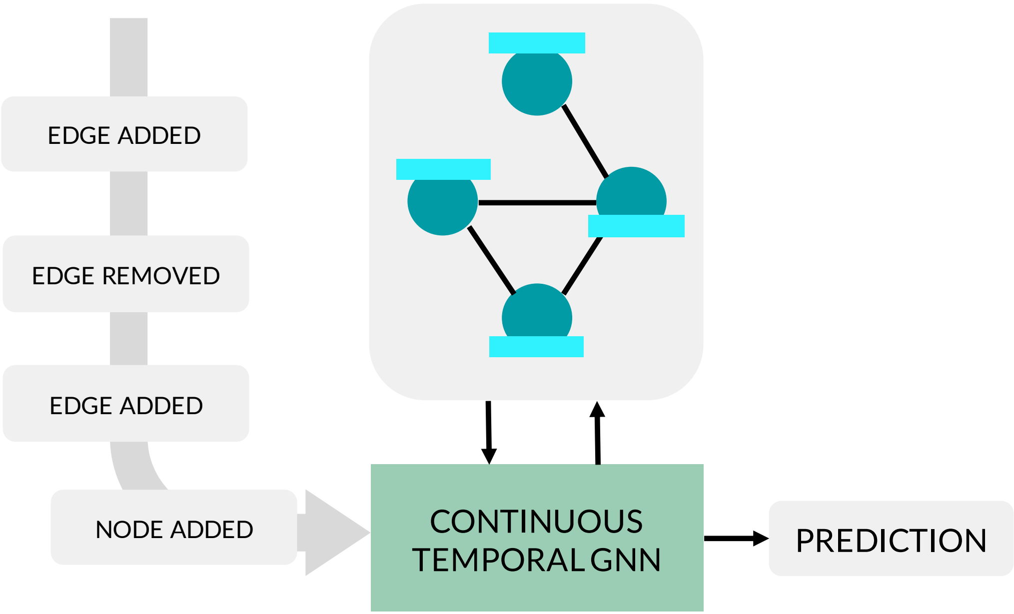


Idea: process **graph snapshots**

Cons:

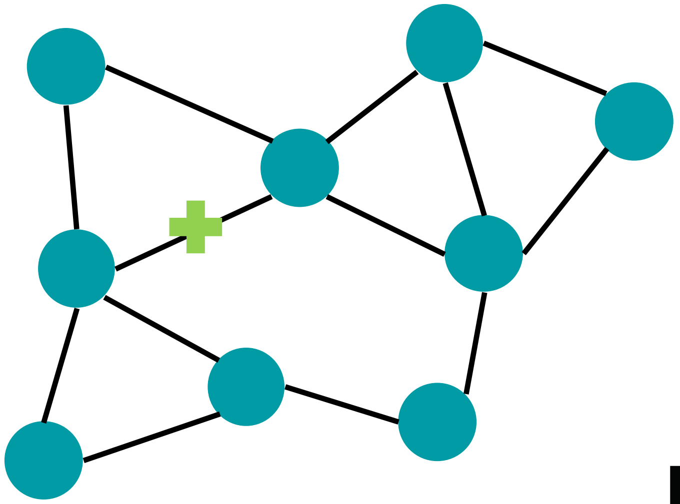
- Full GNN (re-)computation is very costly for large graphs
- High-latency, infrequent predictions

Continuous Temporal GNNs



Process a **stream of graph changes**

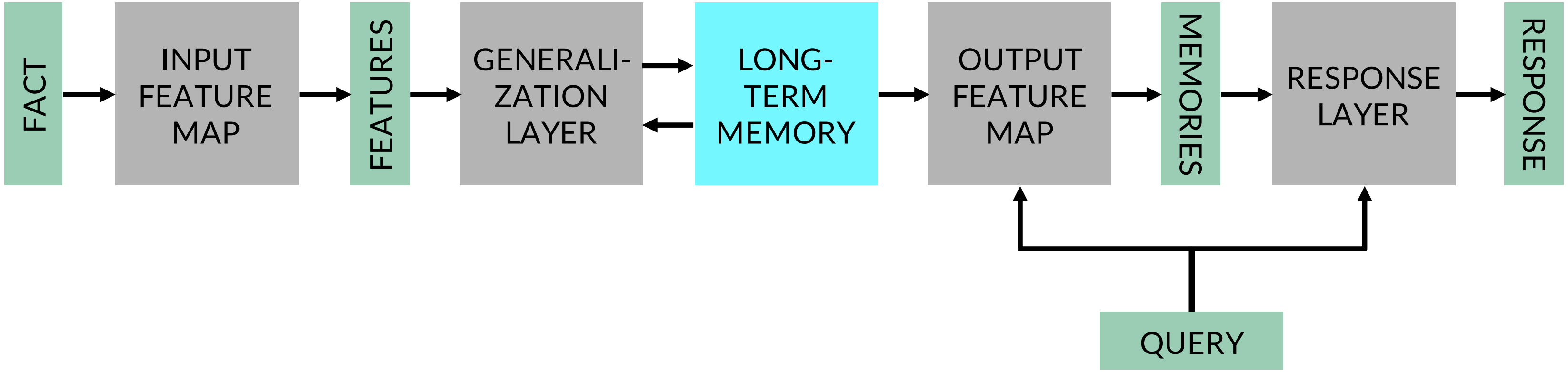
Incremental embedding updates



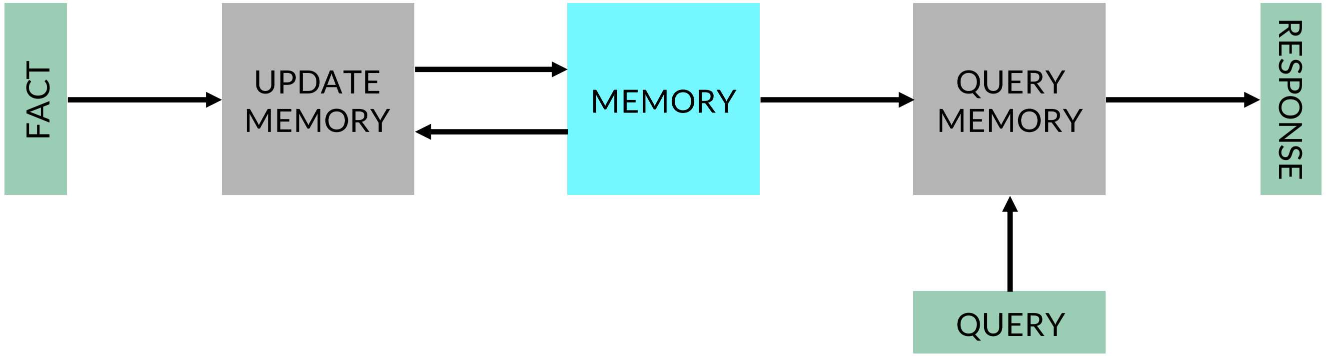
Memory Networks

Introduced by Weston et al. in “Memory Networks”, ICLR 2015

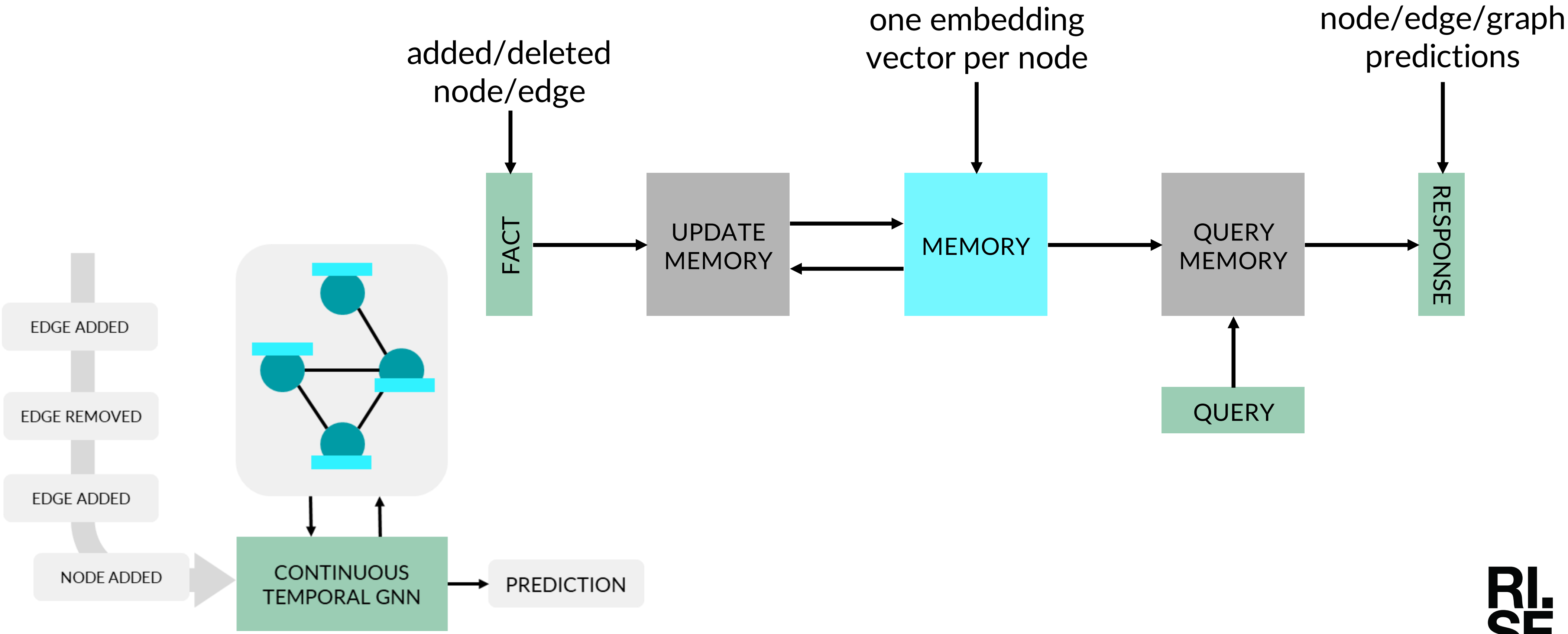
Key idea: teach the models how to **read from and write to a persistent, long-term memory**



Memory Networks (Simplified)

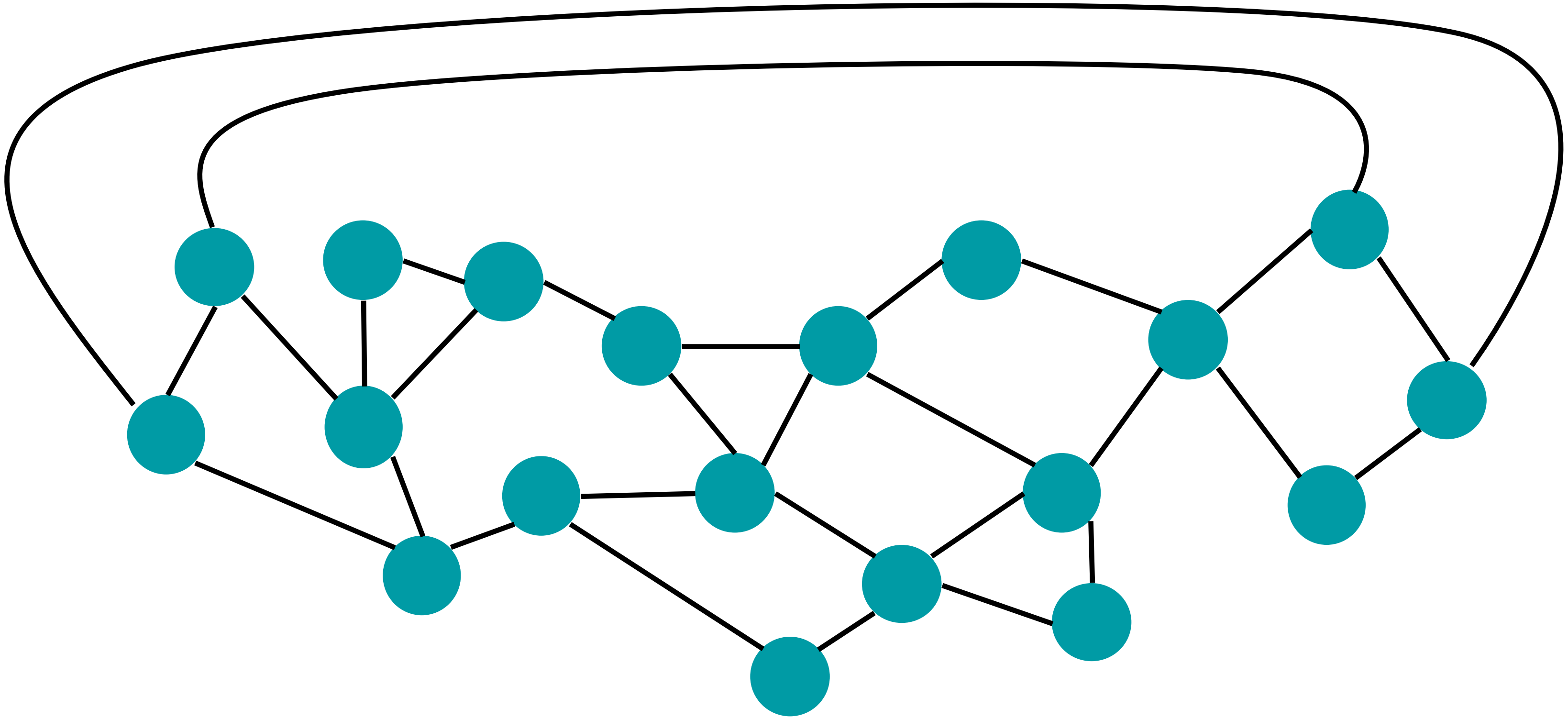


Continuous Temporal GNNs are Memory Networks



The Problem with Incremental Changes

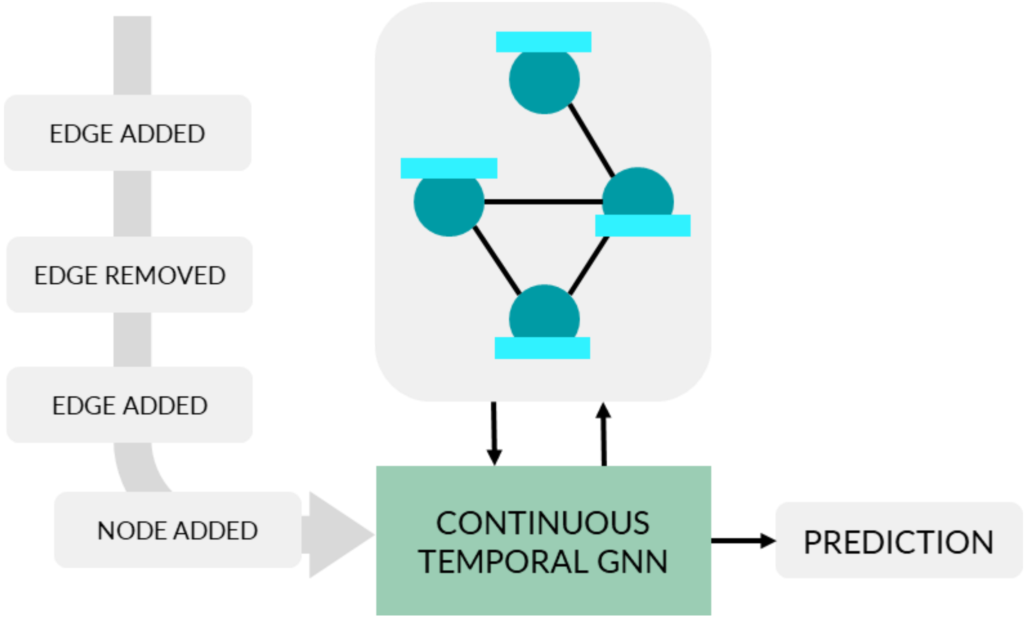
A single edge addition can cause significant changes to the overall structure of a graph



The Problem with Incremental Changes

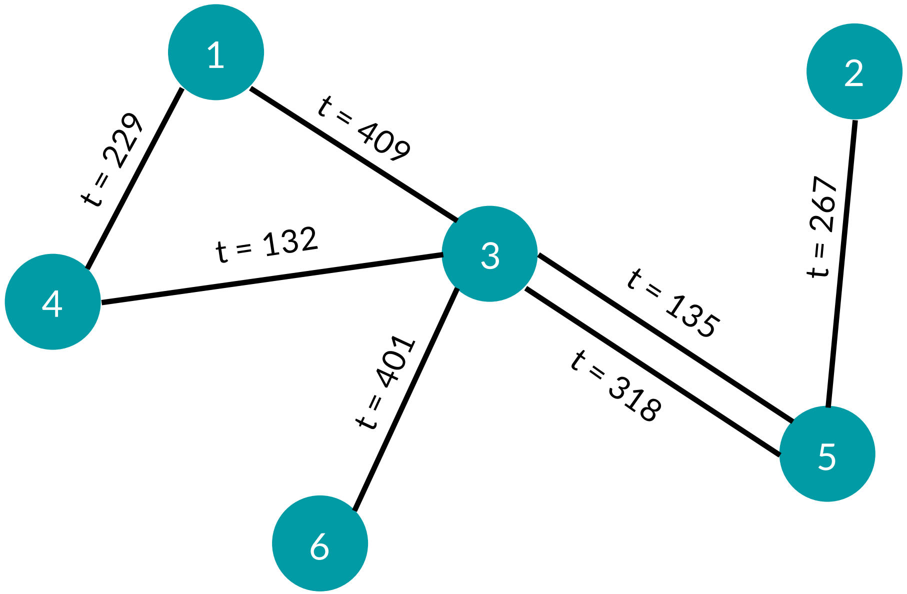
Cons of Continuous Temporal GNNs

- Not scalable to **denser graphs**
- Cannot capture **multi-hop dependencies** in a scalable way



Temporal Interaction Networks

Edges represent **timestamped, instantaneous interactions**, rather than continuous connections

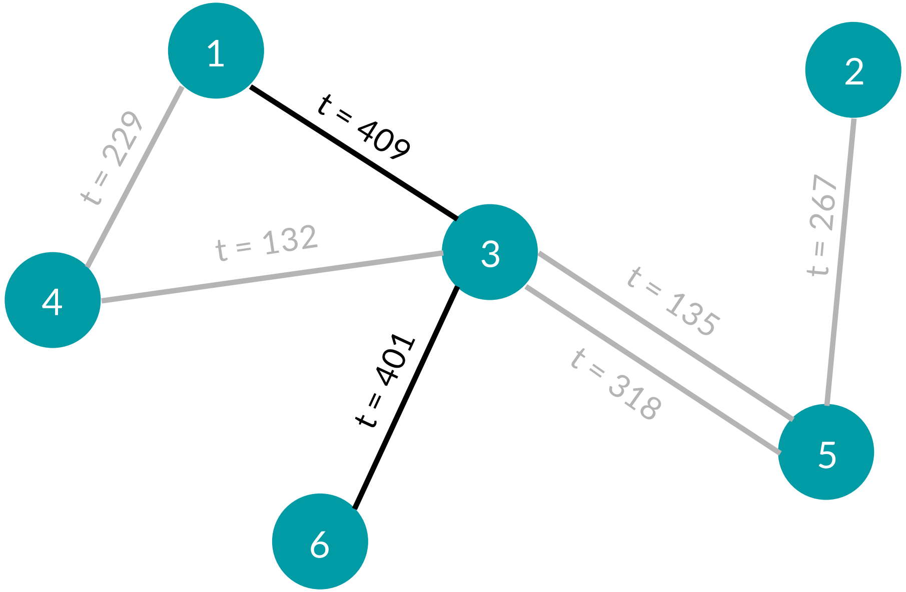


Causal Temporal GNNs

Key insight: past interactions are **not affected by future interactions**

$$h_1^{(409)} = f\left(h_1^{(229)}, h_3^{(401)}\right)$$

$$h_3^{(409)} = f\left(h_3^{(401)}, h_1^{(229)}\right)$$



Fast and scalable!

Requires only the latest embeddings of the interacting nodes

Use Case:
IoT Botnet Detection with Lightweight Memory Networks

RI.
SE

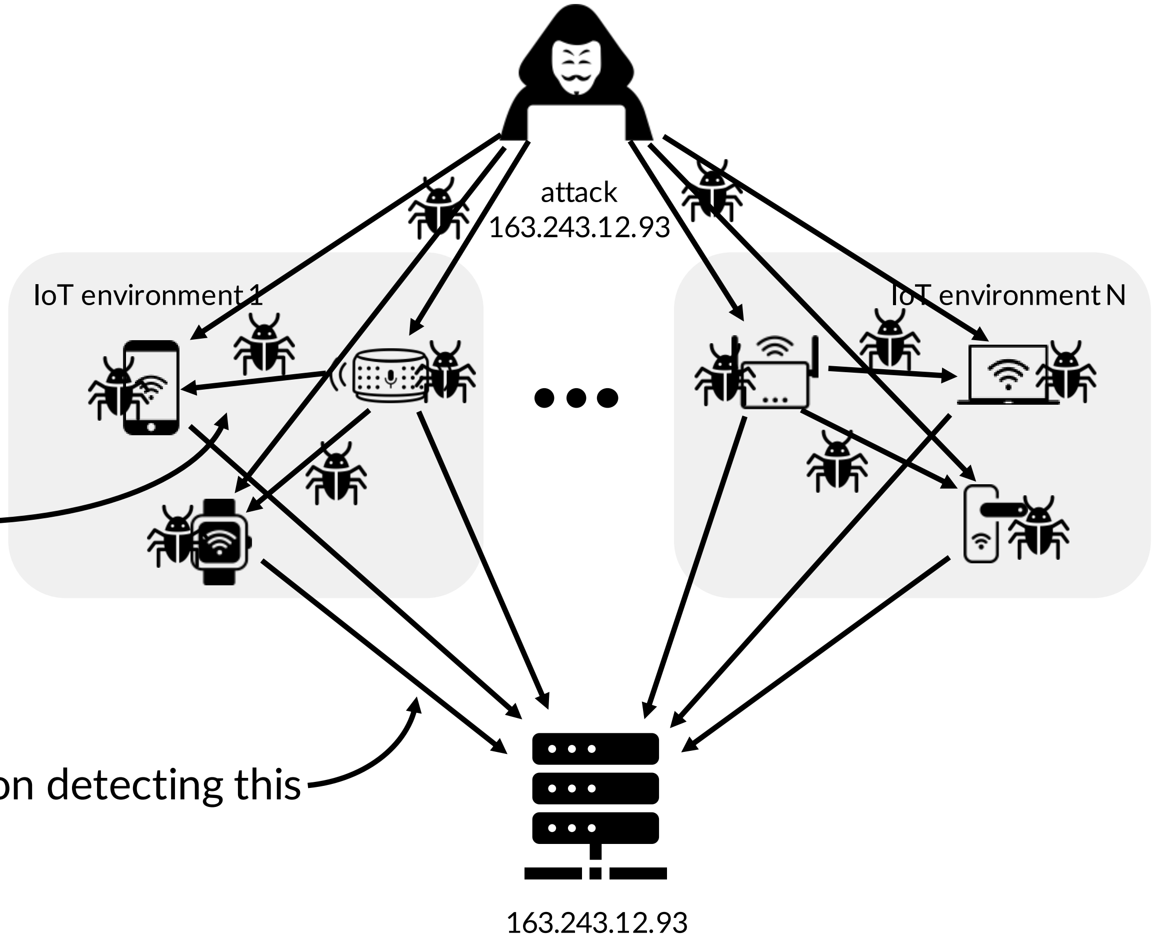
Use Case: IoT Botnet Detection

- Growing number of IoT devices: 30.9 billions in 2025[1]
- IoT security practices are not well-established
- IoT botnets spread easily over the Internet
- IoT botnets are responsible for frequent, large Distributed Denial of Service (DDoS) attacks
  - Infamous Mirai example: 600k infected devices, 1.2 Tbps of malicious traffic[2]
  - Can take down major online services (e.g. DNS resolvers)

1. https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/
2. https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/

IoT Botnets 101

Goal: early-stage botnet detection during the spreading phase



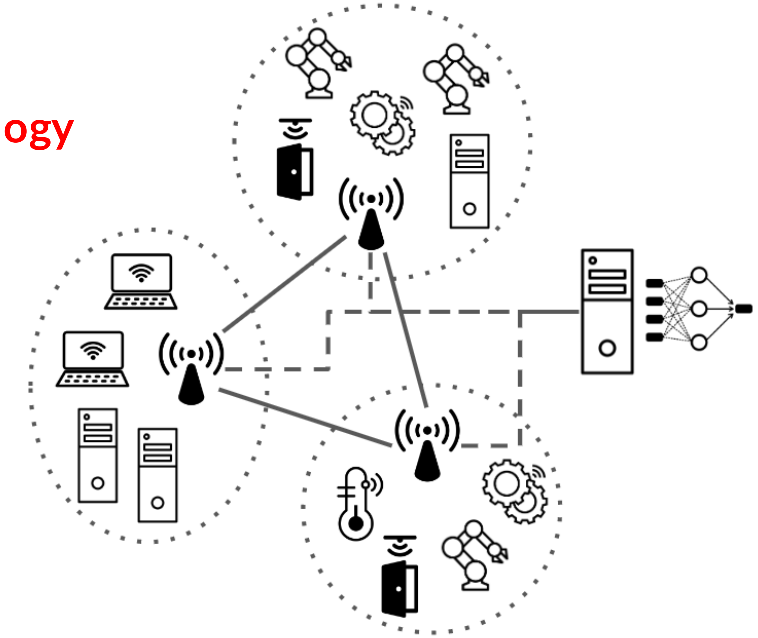
Not so much on this

Lots of work on detecting this

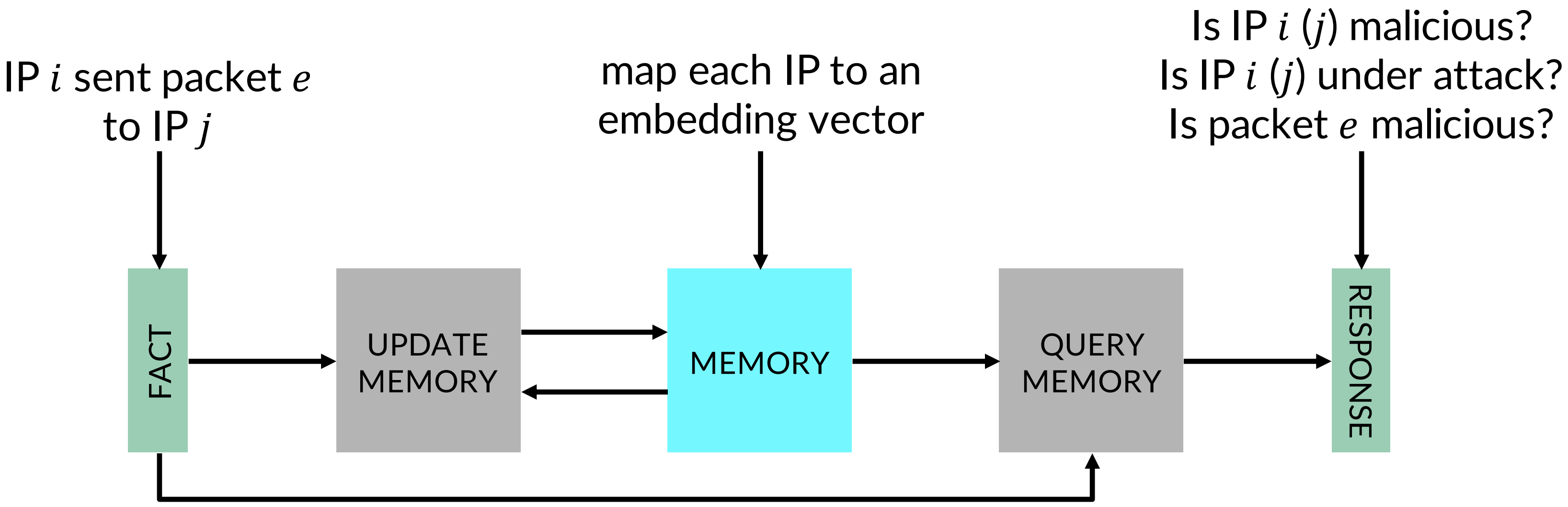
Our Approach

Exploit global knowledge of the dynamic communication network between devices

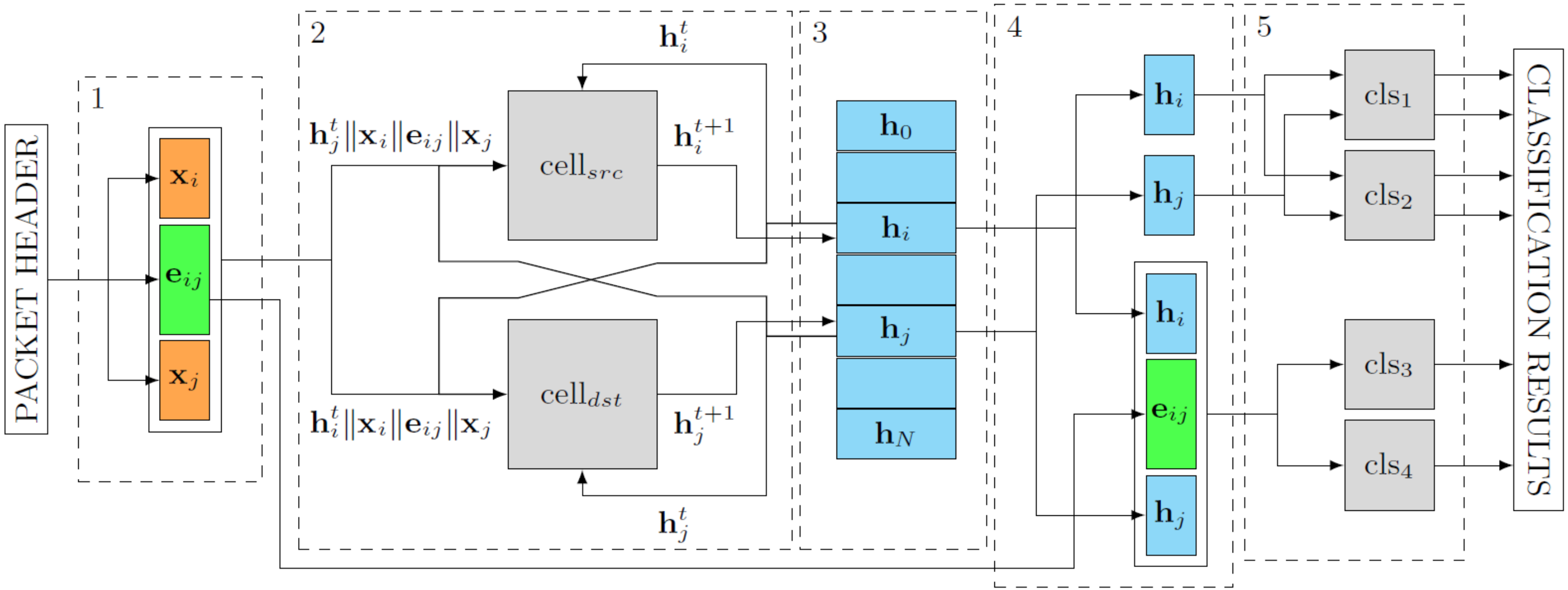
- Build a “profile” for each device
  - Updated for each packet sent/received
  - Take into consideration **the network history and topology**
- **Real-time** detection
  - Small and fast GRL model
- Key insight: **causality!**



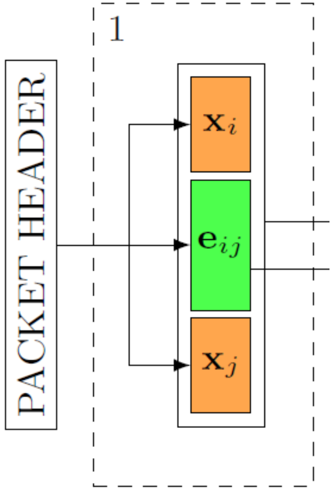
LiMNet: Lightweight Memory Network

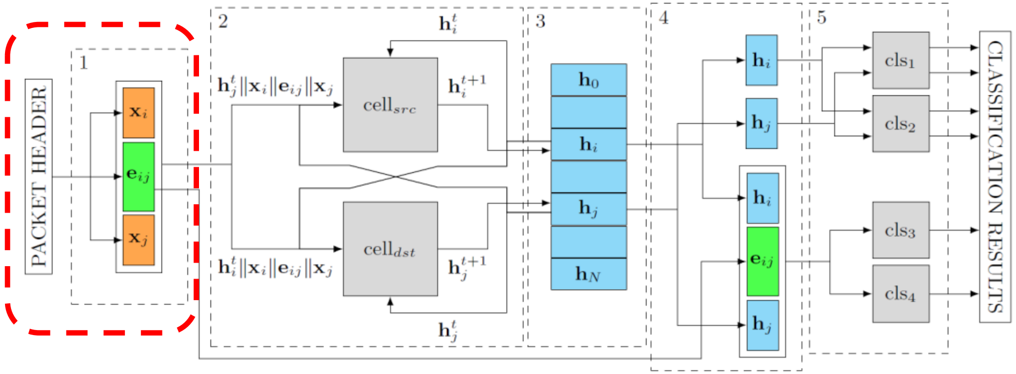


LiMNet Architecture



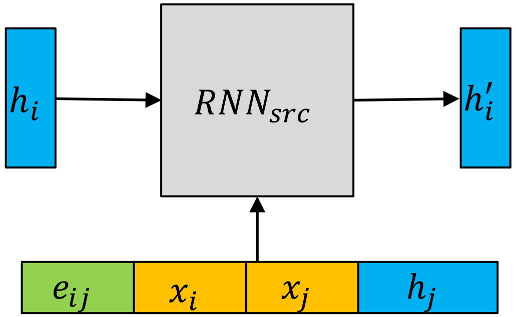
Input Feature Map

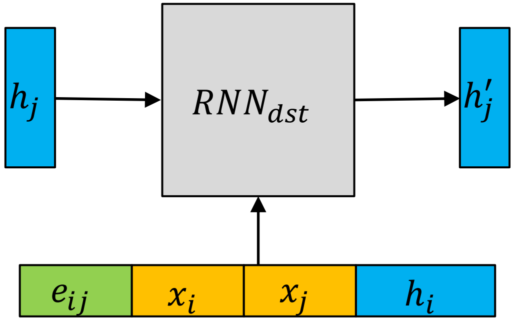


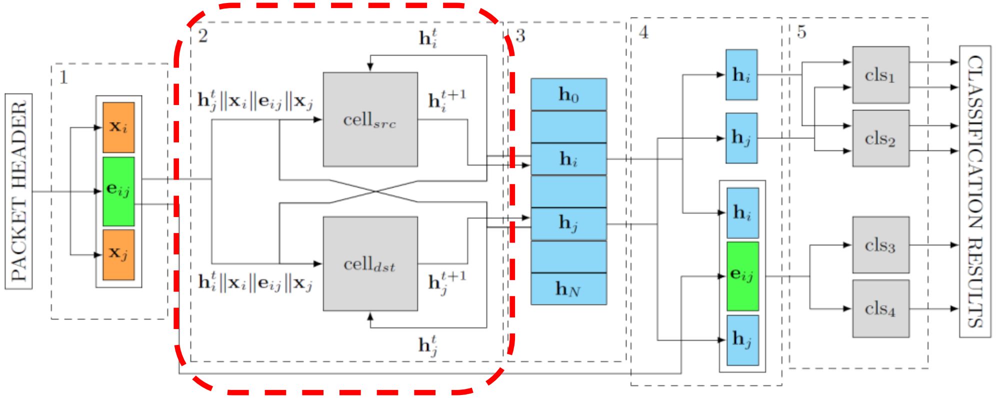


- Source/Destination IP feaures:
  - private vs public IP
  - unicast vs multicast IP
- Packet features:
  - length
  - application/transport protocol

Generalization Layer

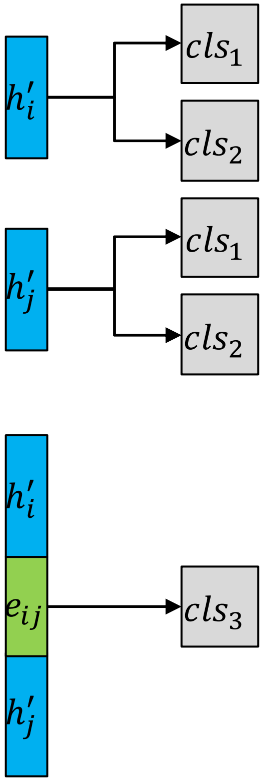


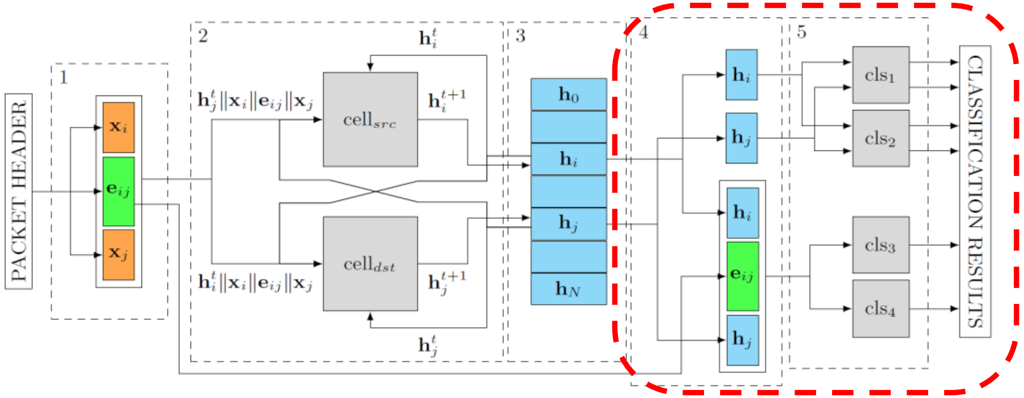




Mutually-recurrent RNN units

Output Feature Map + Response Layer





- **Multi-task learning** with both **node- and edge-level** tasks
  - Identify malicious nodes -> node-level
  - Identify under-attack nodes -> node-level
  - Identify malicious packets -> edge-level
- **Shallow classifiers**

Results

Significant improvement over state of the art methods

Type	Layers	Layer size	Cell type	Device malicious [AUROC]	Device attacked [AUROC]	Packet malicious [AUROC]
recurrent	1	64	LSTM	85.83	97.38	81.04
recurrent	3	32	GRU	85.82	97.52	81.23
LiMNet	1	32	GRU	<u>98.73</u>	<u>98.72</u>	<u>99.72</u>
LiMNet	1	64	GRU	**99.13**	**98.84**	**99.75**

Results

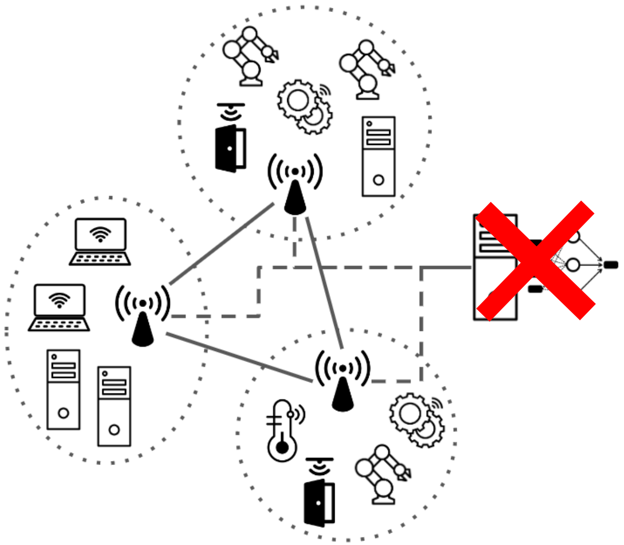
- **Small model**
  - Can fit in the L2 cache of a modern CPU core
- **Fast inference**
  - single CPU core, no accelerators
  - one packet at a time, no batching

Type	Layers	Layer size	Cell type	Model size [KiB]	Inference speed [packets/s]
recurrent	1	64	LSTM	9309	1814
recurrent	3	32	GRU	9472	972
LiMNet	1	32	GRU	**65**	**3381**
LiMNet	1	64	GRU	226	3037

Towards Decentralized Inference

The Centralization Problem

- **Scalability**
  - Large **volume** and **velocity** of graph updates
  - Network is typically the first bottleneck
- **Reliability**
- **Governance**



Goal: decentralized continuous inference on dynamic graphs

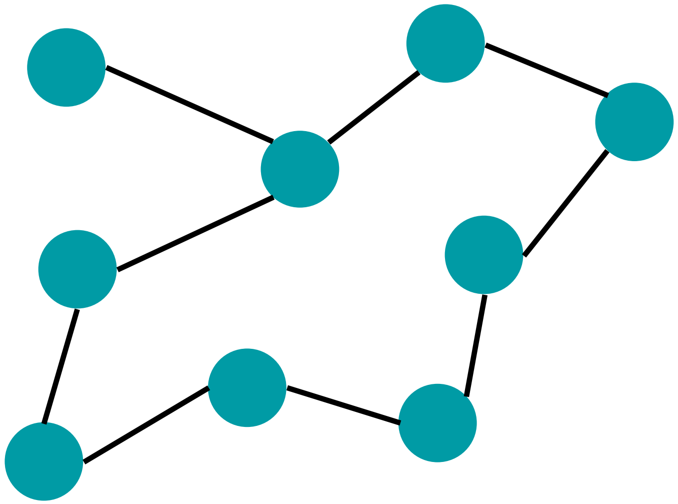
Gossip Protocols

Family of **decentralized, peer-to-peer** protocols

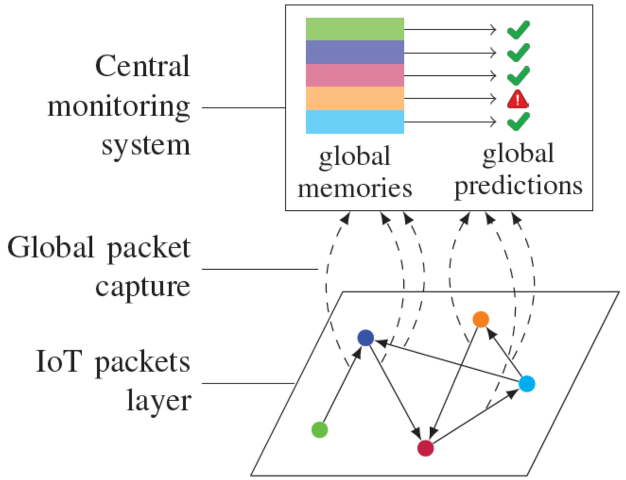
Used for **information dissemination or aggregation**

Key principle: periodic information exchanges with random peers

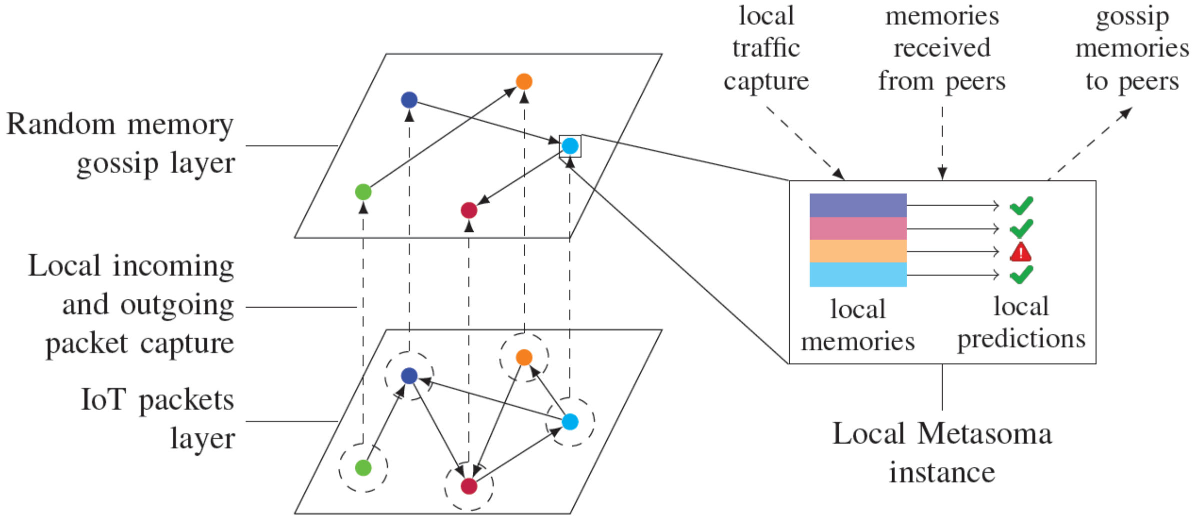
Very efficient!



Decentralized Memory Network

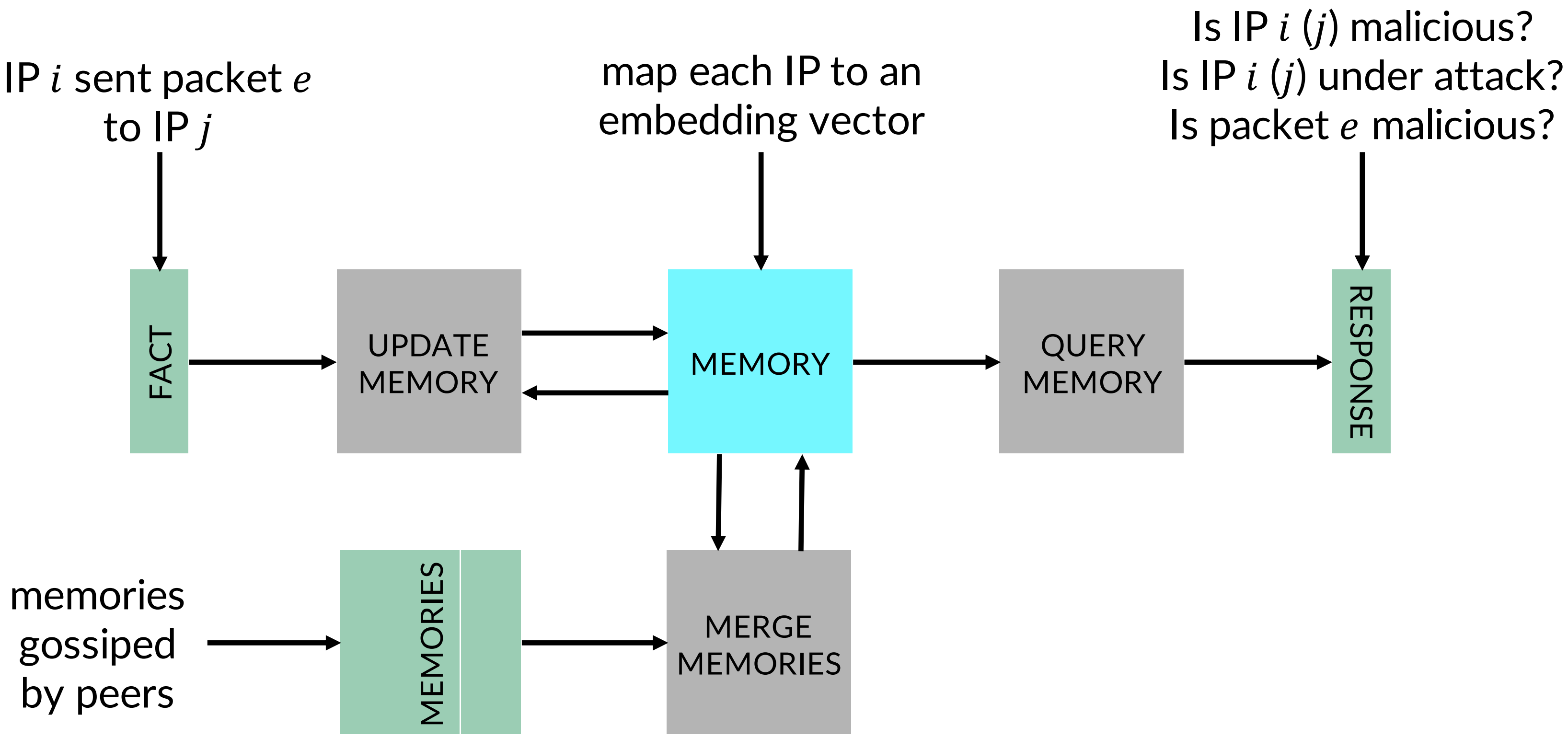


Centralized (e.g. LiMNet)



Decentralized

Metasoma Architecture



Challenges

- **Performance tradeoff**
  – decentralized inference based on partial knowledge will *never* match centralized inference on global knowledge
- **Resource efficiency**
  – Significant overhead on low-power IoT devices
- **Security**
  – Significant increase in the available **attack surface** for malware
  – Metasoma required a **deep security analysis** and complex countermeasures

<u>We do not have a perfect solution, but a promising starting point for further research!</u>

RI.
SE

Conclusion

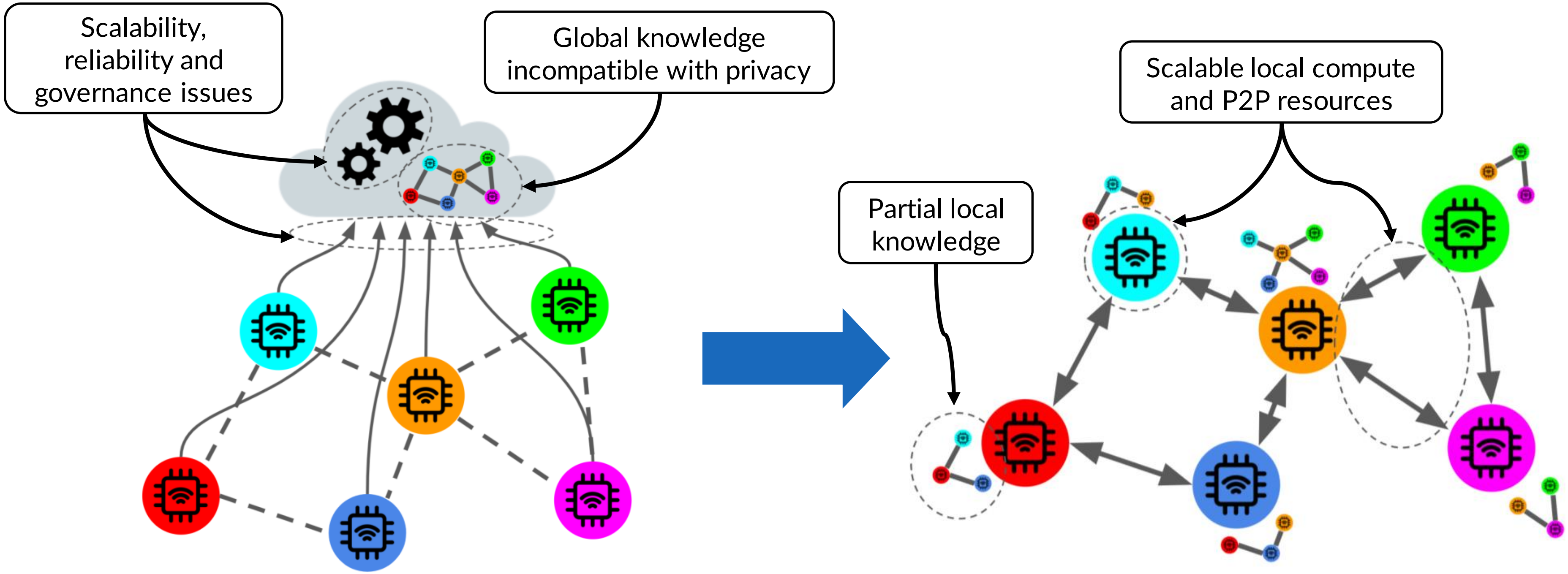
Takeaways

- Graph are everywhere -> **Graph Representation Learning** is key
  - Dynamic graphs -> **Temporal GNNs**
- **Memory networks** -> powerful abstraction
- Temporal Interaction Networks -> **Causal Temporal GNNs**

Our Wider Vision

Centralized Architecture

Decentralized Architecture



References

- Memory networks: Weston et al., *Memory Networks*, ICLR 2015
- LiMNet: Giaretta et al., LiMNet: *Early-Stage Detection of IoT Botnets with Lightweight Memory Networks*, ESORICS 2021
- Metasoma: Giaretta et al., *Metasoma: Decentralized and Collaborative Early-Stage Detection of IoT Botnets*, preprint available https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-325436

My Ph.D. dissertation:

Lodovico Giaretta, *Towards Decentralized Graph Learning*

Acknowledgements

rais-itn.eu

aiotwin.eu

Thank You!
Any Questions?