LINGUAE-ML SEMINAR

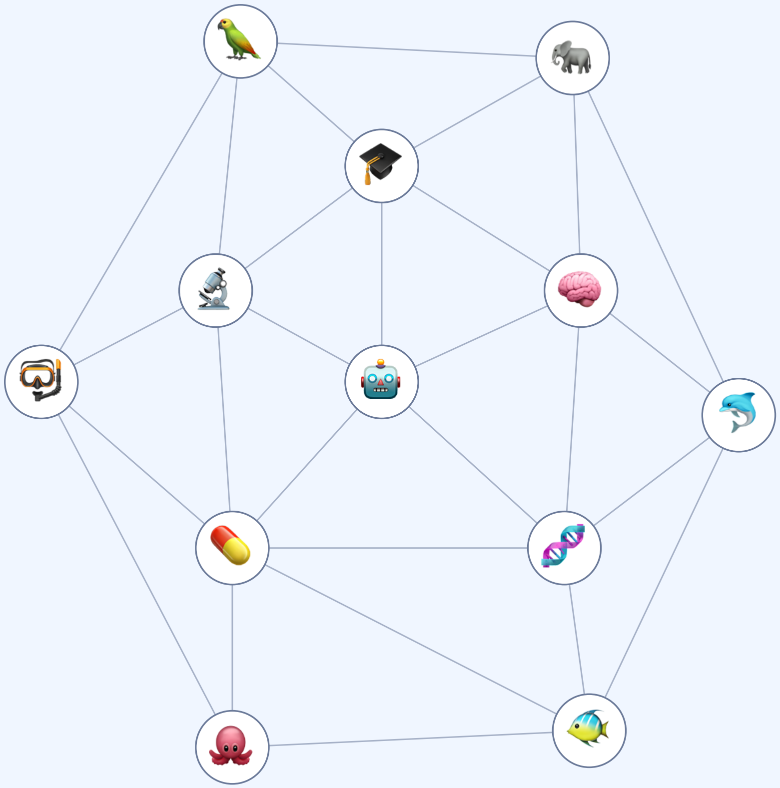
Machine Learning on *Temporal Graphs*

From Foundations to Emerging Frontiers

19/05/2026

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

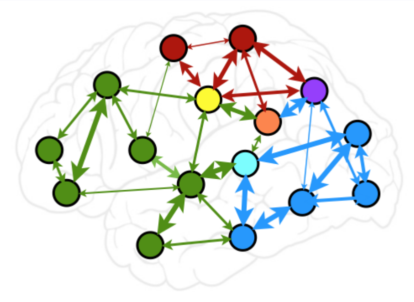


SECTION 01

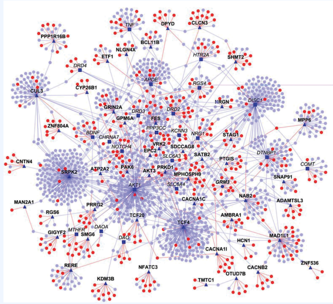
Why Should We Care About ML on Graphs?

Networks are everywhere

And graphs are a great way to model them

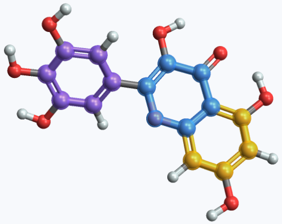


Functional Networks



Interaction Networks

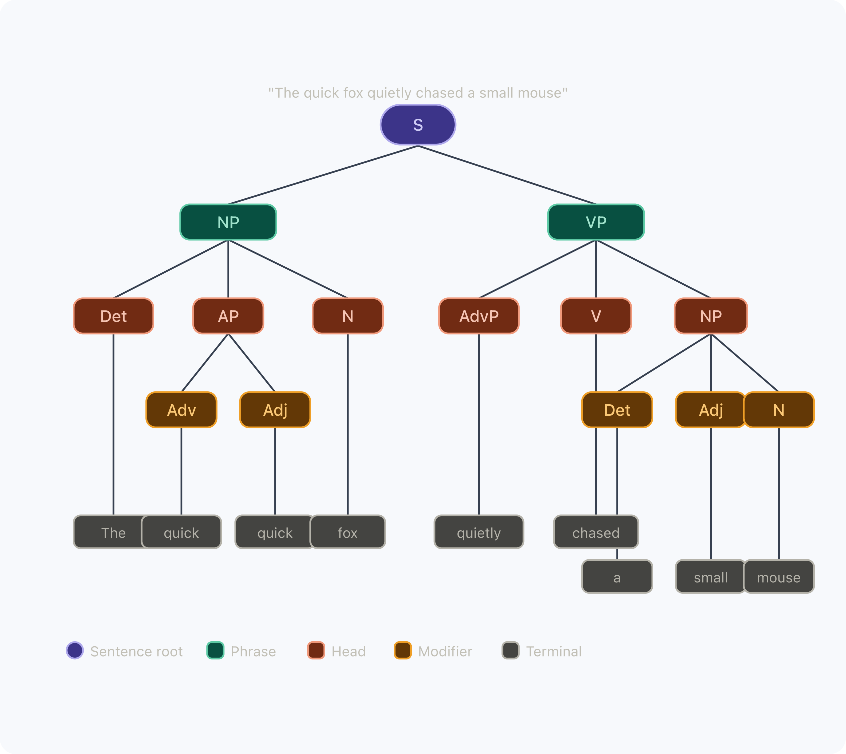
Social Networks



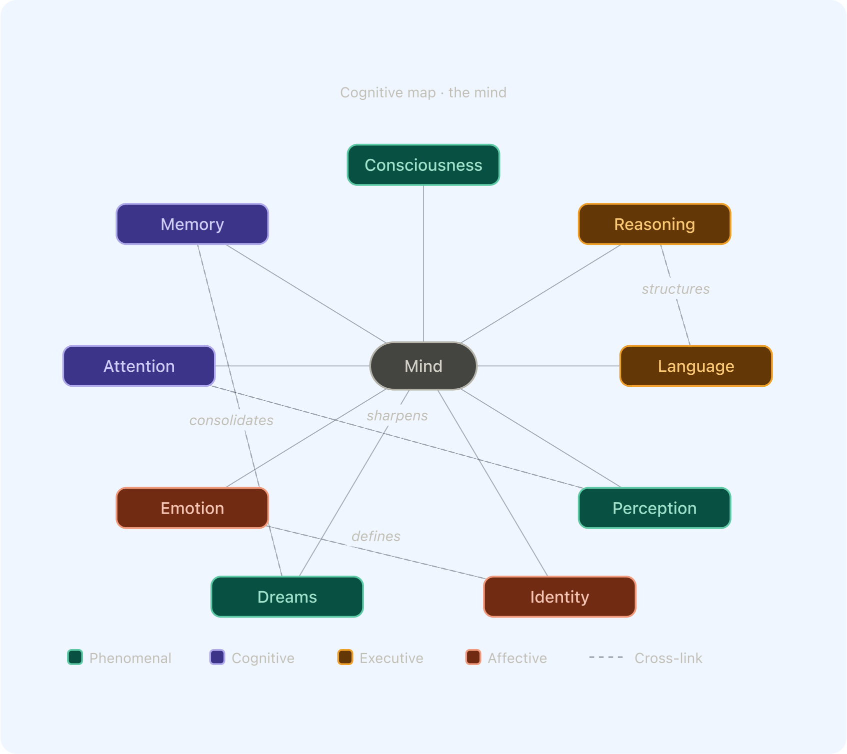
Molecules

Networks are everywhere

And graphs are a great way to model them

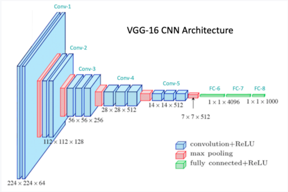
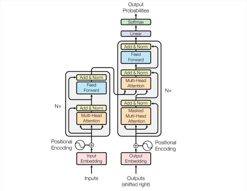


Syntax Trees



Cognitive Maps

What model should we use for graphs?

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

SECTION 02

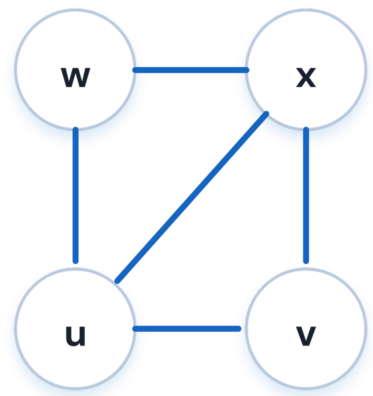
(Static) Graphs and Graph Tasks

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

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

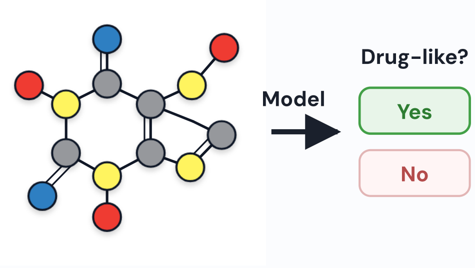
X = node features

	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 classification

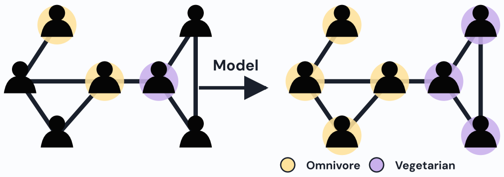
Predict a label for the whole graph.



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

Node classification

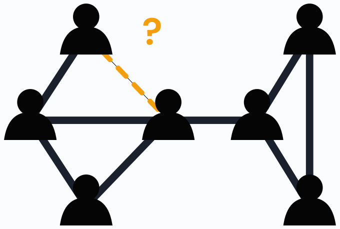
Predict labels for individual nodes using graph context.



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

Link prediction

Score missing or future edges between pairs of nodes.



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

SECTION 03

(Static) Graph Neural Networks

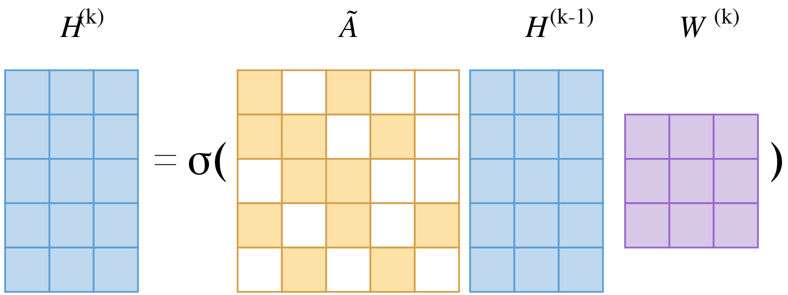
Graph Neural Networks (GNNs)

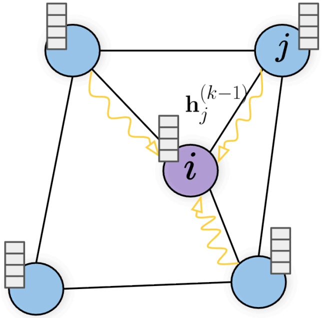
Convolutional GNN

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

Message-Passing GNN

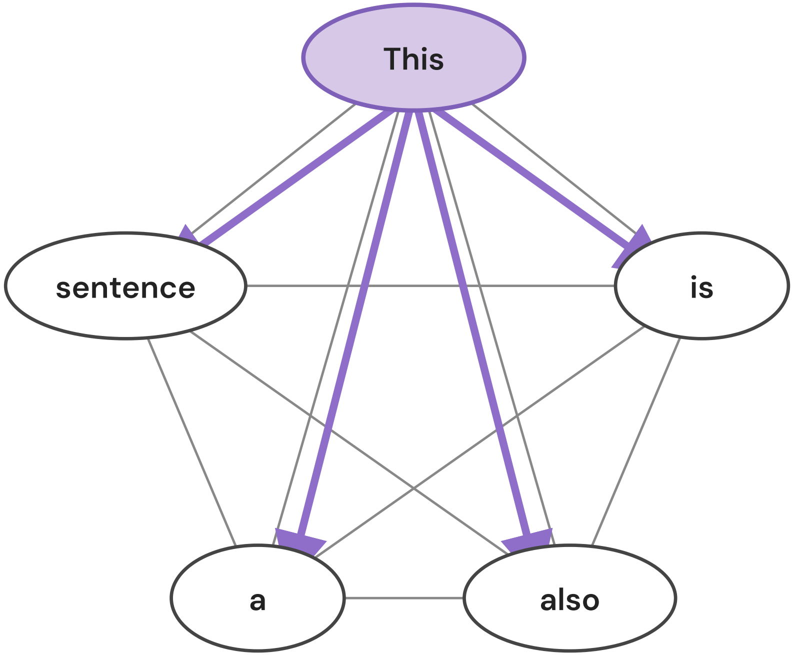
$$\mathbf{m}_i^{(k)} = \mathrm{AGG}^{(k)}\left(\{\{\mathbf{h}_j^{(k-1)} : (i,j) \in E\}\}\right)$$
$$\mathbf{h}_i^{(k)} = \mathrm{COM}^{(k)}\left(\mathbf{h}_i^{(k-1)}, \mathbf{m}_i^{(k)}\right)$$
$$\mathbf{h}_i^{(0)} = \mathbf{x}_i$$





Transformers are GNNs

On the fully connected graph



[9] C. K. Joshi, "Transformers are Graph Neural Networks", The Gradient 2020

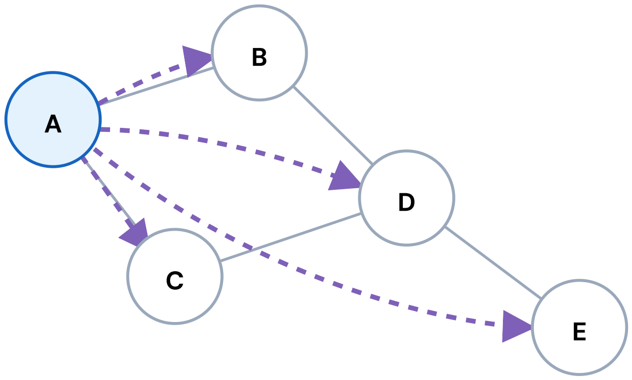
Graph Transformers [10]

Transformer-style attention, but with graph structure injected

CORE IDEA

- Self-attention lets every node aggregate from all other nodes
- Graph information enters through special positional encodings

$$\mathrm{Attn}(i,j) \leftarrow q_i^\top k_j + b_{\mathrm{graph}}(i,j)$$



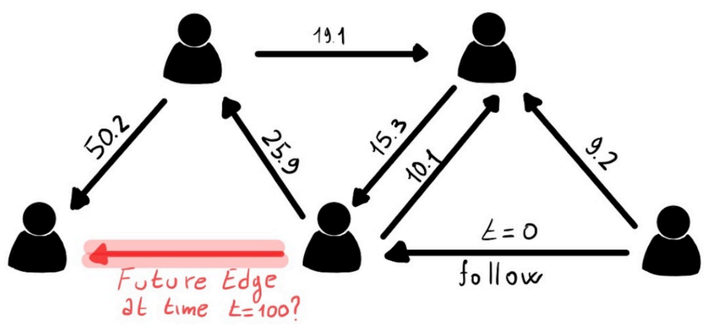
[10] C. Ying et al., "Do Transformers Really Perform Bad for Graph Representation?", NeurIPS 2021

SECTION 04

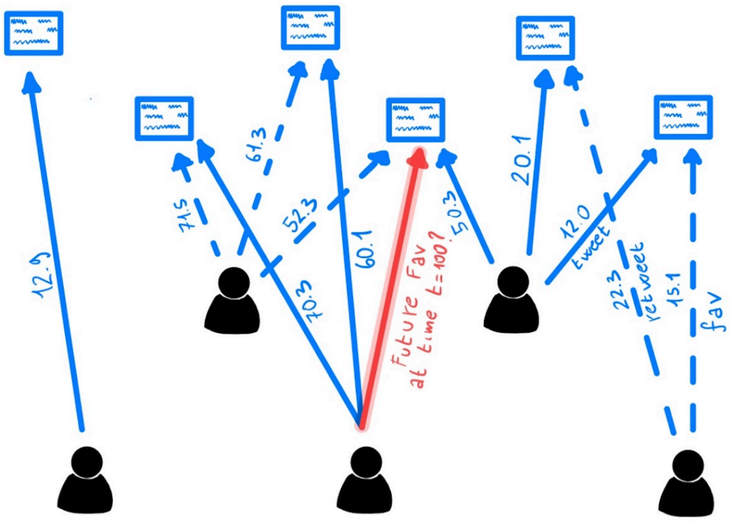
Dynamic Graphs

Some Examples of Dynamic Graphs

Graphs changing over time



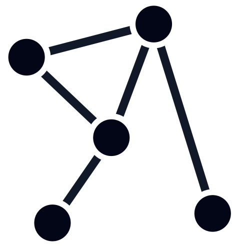
Social Networks



Interaction Networks

From Static to Dynamic Graphs

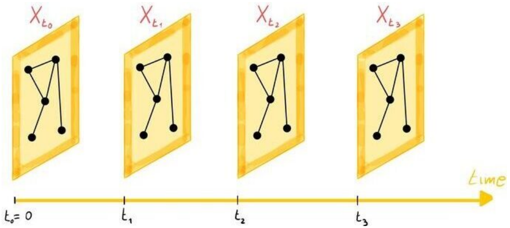
$$G = (V, E, X)$$



Static Graph

- No notion of time
- Examples: molecules, syntax trees

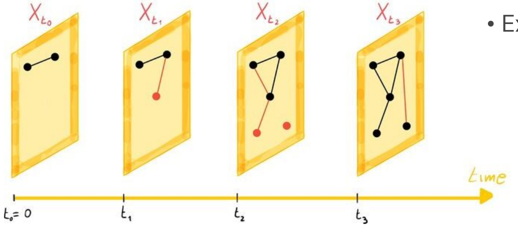
$$G_t = (V, E, X_t)$$



Spatio-Temporal Graph

- Fixed topology; changing features
- Regular time intervals
- Examples: traffic, weather sensors

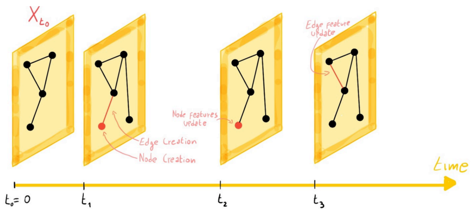
$$G_t = (V_t, E_t, X_t)$$



Discrete-Time DG (DTDG)

- Changing topology and features
- Regular time intervals -> sequence of snapshots
- Examples: weekly trade networks

$$G(t) = \{x_{t_1}, x_{t_2}, \dots\} \quad t_1 \leq t_2 \leq \dots$$



Continuous-Time DG (CTDG)

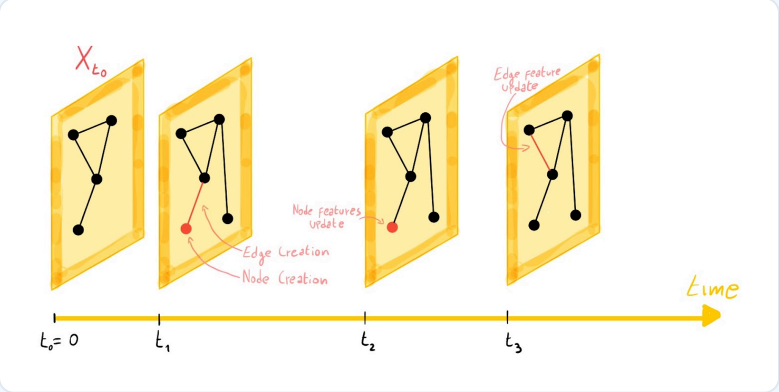
- Most general formulation
- Sequence of timestamped events
- Examples: social, financial

Less General → *More General*

Why Learning on Dynamic Graphs is Different

Temporal graph models must exploit the event history without collapsing it into one static snapshot

WHAT CHANGES



- Events arrive in order: creation, deletion, update, interaction
- The model should use both *what* happened and *when* it happened
- New tasks ask what happens next, or when it happens

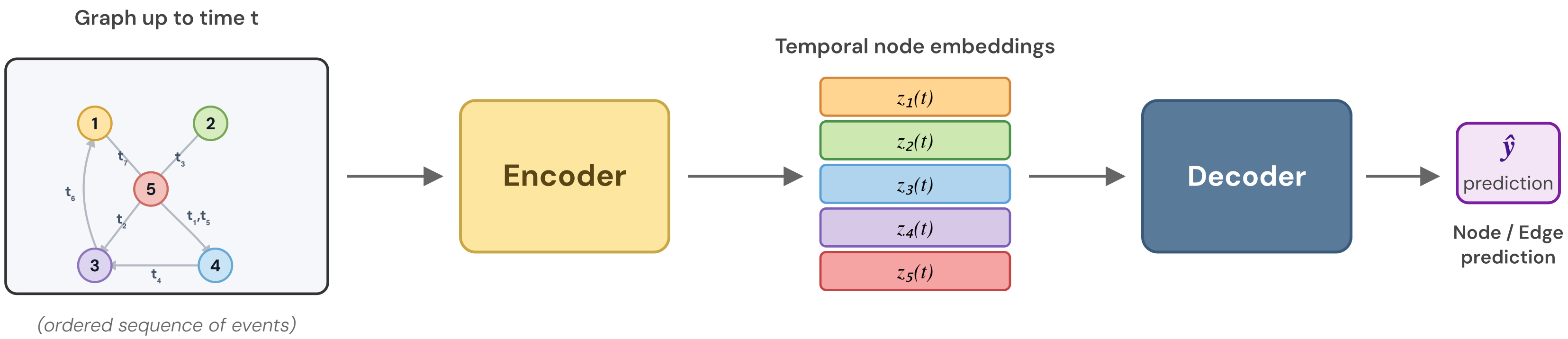
WHY STATIC GNNS ARE NOT ENOUGH

- *Information loss*: the last snapshot hides the evolution path
- *Inefficiency*: every new event can force repeated computation
- *No timing prediction*: static GNNs do not support predicting when something will happen

SECTION 05

Models

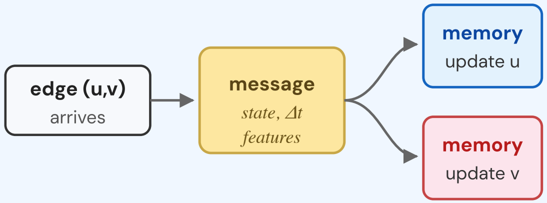
Temporal Graph Model



Memory-based Models [11, 12]

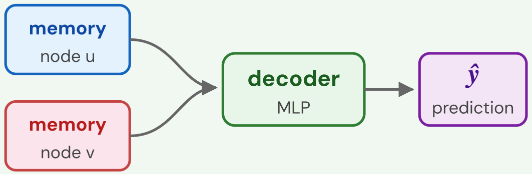
- Process events in order using an RNN, with a different hidden state per node
- Hidden state (memory) is a compressed representation of all past interactions of a node
- Memory is directly used as the temporal node embeddings

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

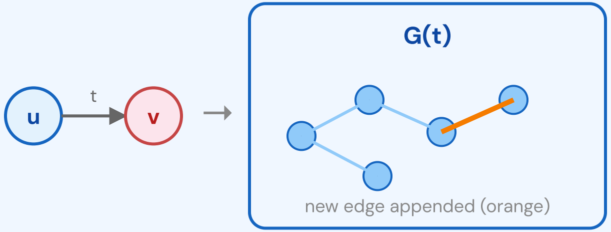
CONS

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

Graph-based Models [13]

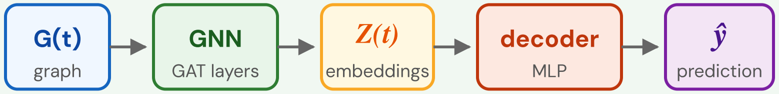
GNN on the graph of previous interactions, with timestamps as edge features

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**

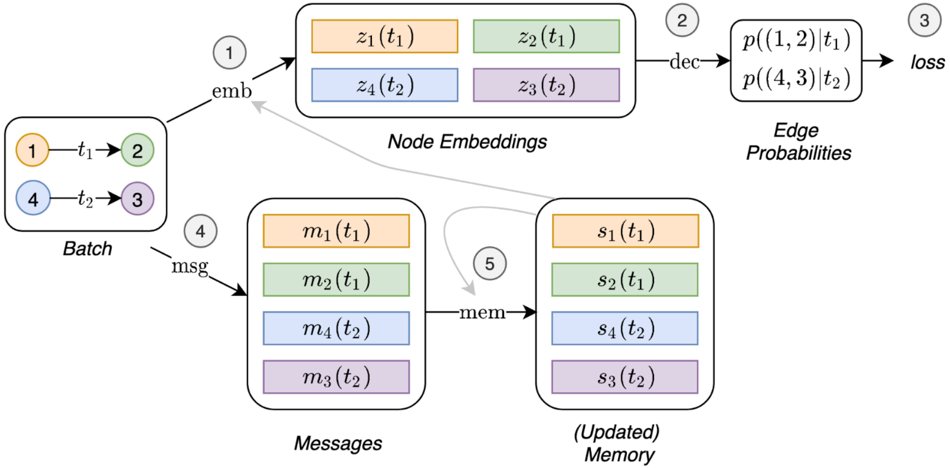
[13] D. Xu et al., "Inductive Representation Learning on Temporal Graphs", ICLR 2020

TGN: Temporal Graph Networks [14]

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

TGN CONTRIBUTION

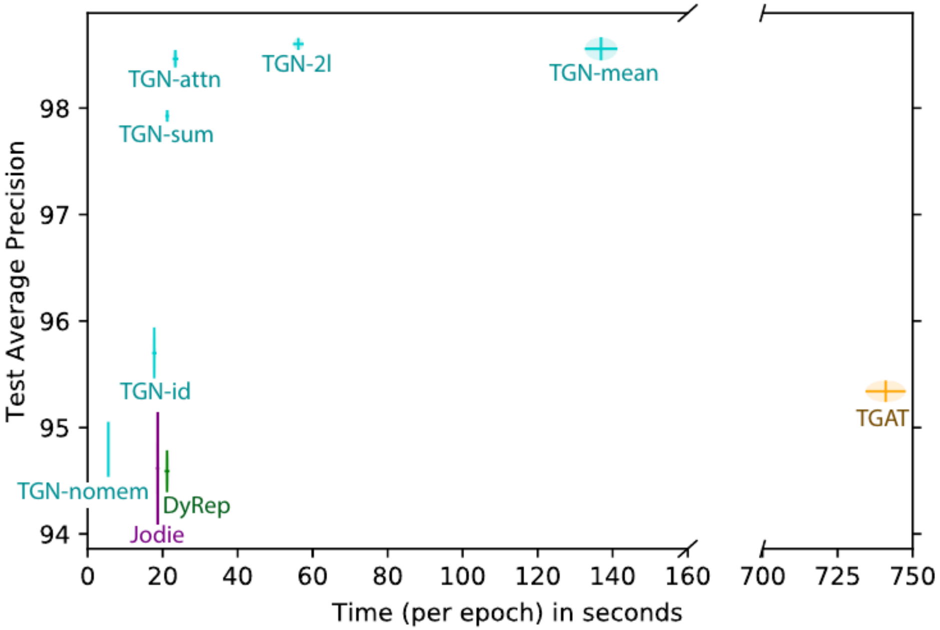
- General framework combining the best of memory-based and graph-based approaches
- Updates node memories from events, then uses a GNN over the interaction graph to produce embeddings
- Generalizes previous memory-based models and graph-based models in one notation



[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

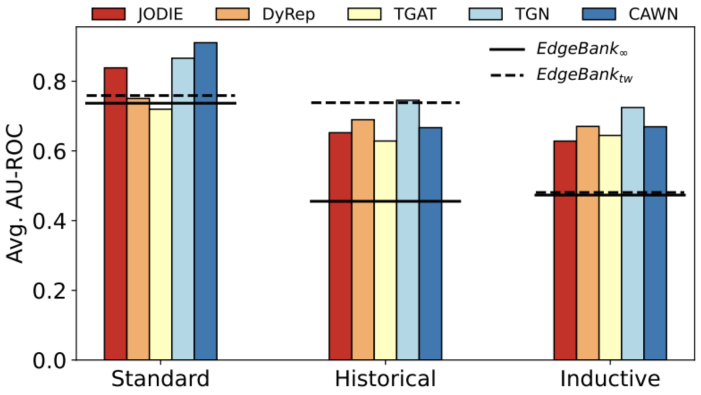
High accuracy with much lower per-epoch cost than the previous methods



[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

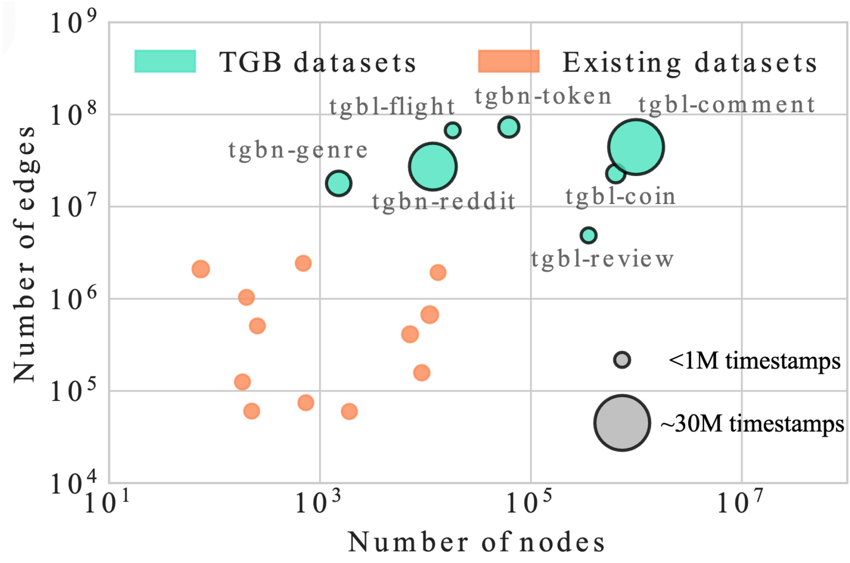
[15] F. Poursafaei et al., "Towards Better Evaluation for Dynamic Link Prediction", NeurIPS Datasets and Benchmarks Track 2022 · [16] L. Yi et al., "TGB-Seq Benchmark: Challenging Temporal GNNs with Complex Sequential Dynamics", ICLR 2025

SECTION 06

Benchmarks

Temporal Graph Benchmark [4]

Diverse, large datasets and unified evaluation



KEY FINDINGS

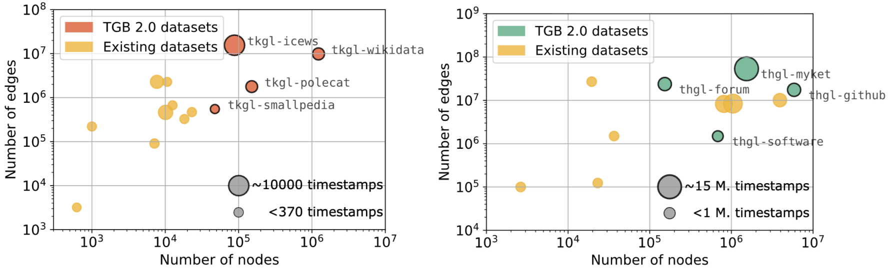
- Previous datasets were too easy and saturated
- Exposed that simple historical baselines can be surprisingly strong
- Large new datasets make scalability part of the benchmark

9 datasets · 5 domains · up to 72M edges

[4] S. Huang et al., "Temporal Graph Benchmark for Machine Learning on Temporal Graphs", NeurIPS Datasets and Benchmarks Track 2023

Temporal Graph Benchmark 2.0 [5]

TGB but for knowledge graphs



(a) Novel Temporal Knowledge Graphs (b) Novel Temporal Heterogeneous Graphs

KEY FINDINGS

- Edge and relation type information is crucial for strong performance
- Simple heuristic baselines remain competitive with more complex methods
- Many methods fail to run on the largest datasets, making scalability a central result

8 datasets · 5 domains · up to 53M edges

[5] J. Gastinger et al., "TGB 2.0: A Benchmark for Learning on Temporal Knowledge Graphs and Heterogeneous Graphs", NeurIPS Datasets and Benchmarks Track 2024

SECTION 07

Unifying Models for Discrete and Continuous Time Graphs

UTG: Unifying Snapshot & Event-Based Models

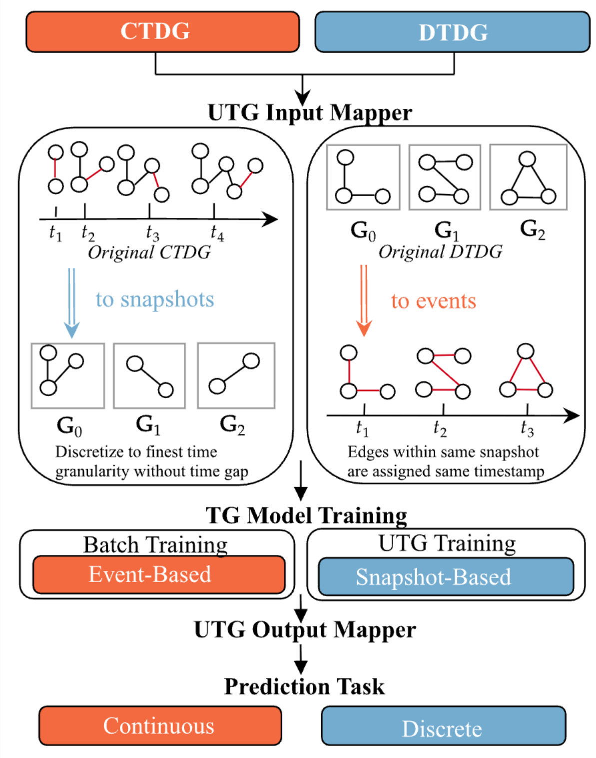
Huang, Poursafaei, Rabbany, Rabusseau, Rossi, LoG 2024

THE PROBLEM

Snapshot (DTDG) and event-based (CTDG) models developed **in isolation**: limited cross-comparison and no unified evaluation.

CONTRIBUTION

- **Input mapper**: convert CTDG to snapshots and DTDG to events, so any model can run on any data
- **UTG training**: use streaming training to make snapshot-based models operate on event streams
- **Output mapper**: align predictions to either continuous-time or discrete-time tasks



UTG: Key Findings

SPEED

Snapshot-based models are **≥ 10× faster** at inference than most event-based models.

PERFORMANCE

With UTG training, snapshot-based models **match TGN & GraphMixer** even on event-based (CTDG) datasets.

INSIGHT

NAT & DyGFormer's edge comes from **joint neighbourhood features**, not from the event-based format: these can be added to snapshot models too.

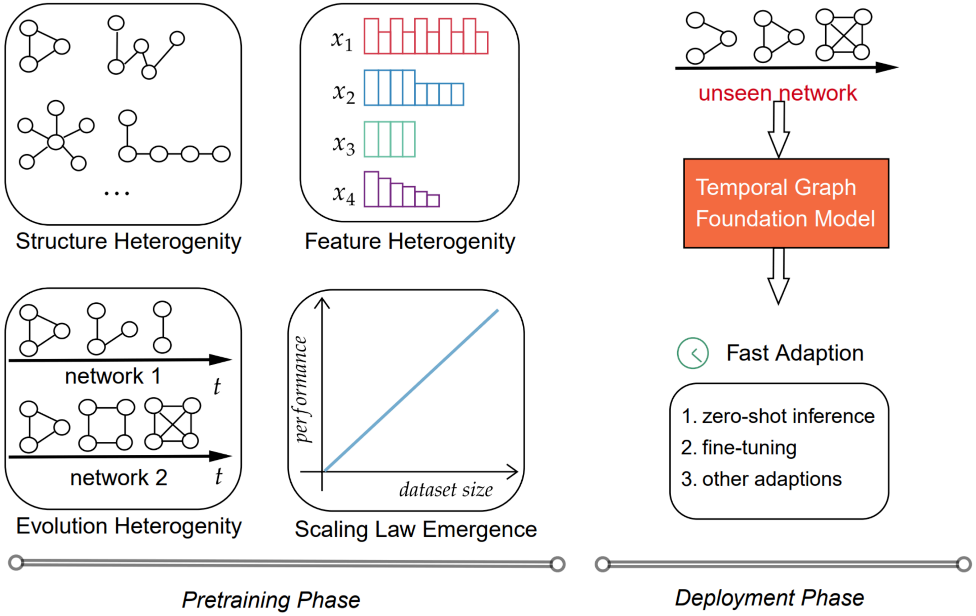
SECTION 08

Next Frontiers

Foundation Models for TG

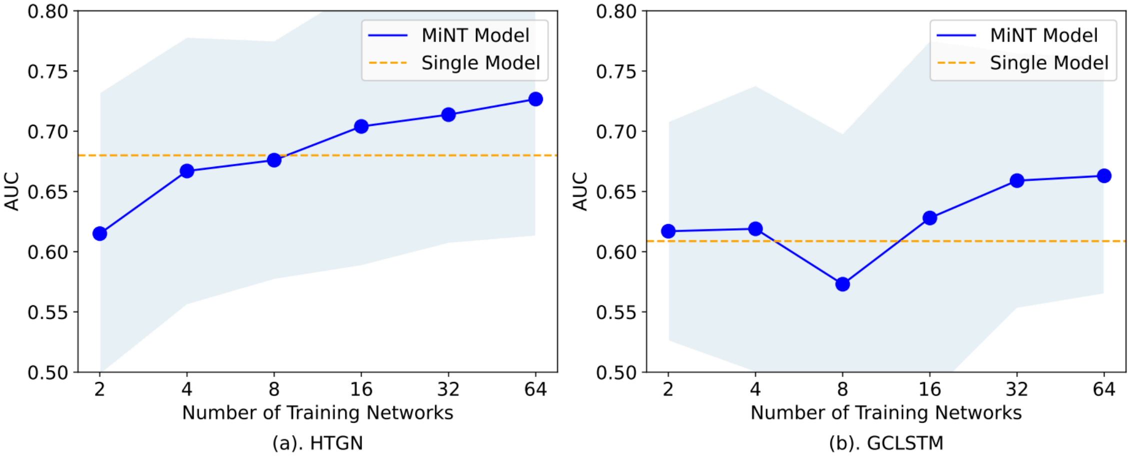
Existing TG models are **trained and tested on the same graph**.

Can we train a single TG model that works on *unseen graphs*, ideally from different domains?



MiNT: Multi-Network Transfer Benchmark for Temporal Graph Learning

- **New dataset:** 84 distinct ERC-20 token transaction networks
- **New framework** to train existing TG models on multiple graphs simultaneously
- **Scaling law for TGs**: model performance improves as it is trained on more networks

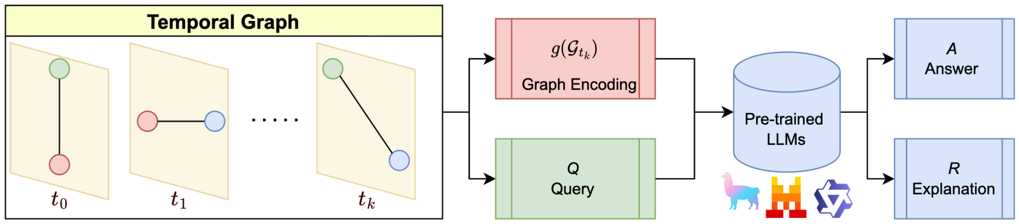


[17] K. Shamsi et al., "MiNT: Multi-Network Transfer Benchmark for Temporal Graph Learning", NeurIPS Datasets and Benchmarks Track 2025

Can we use LMs directly on temporal graphs?

TGTalker

Translate temporal graph structure into natural language and feed it to a **pre-trained LLM** — no fine-tuning required.



Four prompt components

- **Background set** — recent edges as context
- **Example set** — 5-shot Q&A pairs
- **Query set** — target edge to predict
- **Temporal neighbors** — 1-hop recent neighbours

[18] S. Huang et al., "Are Large Language Models Good Temporal Graph Learners?", arXiv 2506.05393 2025

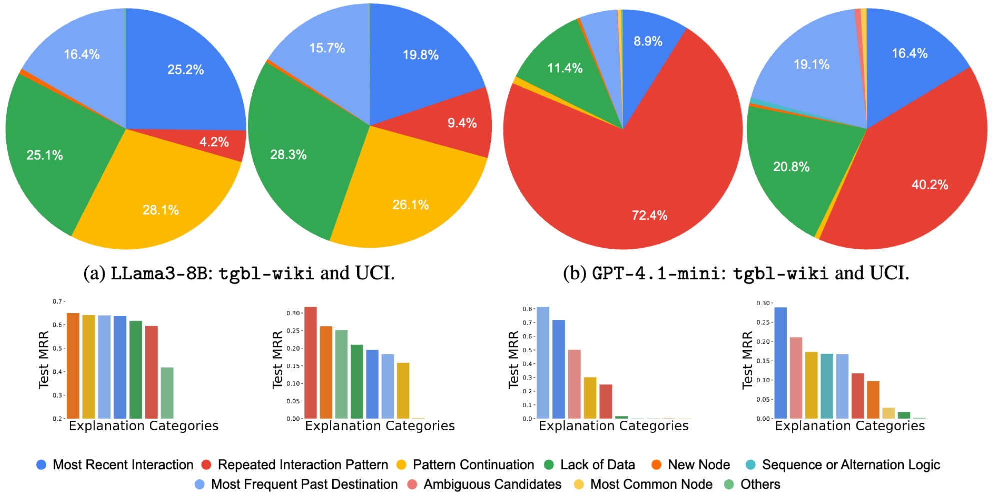
TGTalker: Results & Explainability

RESULTS

- Competitive without fine-tuning
- Consistently outperforms TGN and HTGN

10 EXPLANATION CATEGORIES DISCOVERED

- **Most Recent Interaction** ≈ EdgeBank heuristic
- **Most Frequent Destination** ≈ PopTrack heuristic
- Novel patterns: sequence logic, analogy-based inference



[18] S. Huang et al., "Are Large Language Models Good Temporal Graph Learners?", arXiv 2506.05393 2025

Resources for TGL

Blog posts, talks, surveys, libraries & datasets

📝 BLOG POSTS

[Temporal Graph Learning in 2023]

[Temporal Graph Learning in 2024]

🎙️ TALKS & READING GROUPS

Andy Huang Lecture Series

Temporal Graph Reading Group
recent papers and talks

📚 SURVEY

Graph Neural Networks for temporal graphs: state of the art, open challenges, and opportunities

💻 LIBRARIES

TGM
Temporal Graph Models library

📊 DATASETS

TGB
Temporal Graph Benchmark

TGB-Seq
Sequential Temporal Graph Benchmark

(a small taste of ongoing work)

SECTION 09

My Current Directions

🔊 Communicating Sound Through Natural Language

A sender agent describes a sound; a receiver agent decodes the description back to audio

ORIGINAL AUDIO

0:00 / 0:05

→

TRANSMITTED SENTENCE

"At entry the envelope carries mid-power, punchy, and extreme-oscillation; through sustain and release, the trace returns swift-onset and aggressive..."

→

RECONSTRUCTED AUDIO

0:00 / 0:05

COMPARISON

Method	Human readability	LLM-native transport	Semantic editing	Acoustic interpretabili	Training-free	Generative decoding	Bandwidth efficiency
Lossless codec (FLAC, WAV)	○	○	○	○	●	○	○
Handcrafted descriptors (MFCC, spectral centroid)	○	○	○	●	●	○	●
Neural codec (EnCodec, SoundStream)	○	◐	○	○	○	●	●
Audio-language tokenizers (AudioLM, TASTE)	○	◐	◐	○	○	●	●
Unconstrained text caption	●	●	●	◐	●	●	○
LAC (this paper)	●	●	●	●	●	●	◐

🐬 OpenWhistle Dataset

- Very rich and unique **dolphin dataset** collected over 5 years
- New benchmark for *whistle detection and classification*

Dataset	# Whistles	Voc. hours	Time span (yrs)	Stable pod (# indiv.)	Setting	Seq. context	Open
OpenWhistle Pretraining	~180,000*	114.3	5.0	● (5)	Semi-nat.	●	●
OpenWhistle Expert subset	8,354	1.9	0.42	● (5)	Semi-nat.	○	●
DOLPHINFREE	4,600	7.3	2.0	○	Wild	○	●
Di Nardo et al., 2025	3,111	0.6	0.003	● (7)	Captive	○	●
Watkins MMSD	566	N/R	70+	○	Wild	○	●
Korkmaz et al., 2023	~29,000*	6.8	0.07	○	Semi-nat.	●	◐
Sicily Strait PAM	14,048	N/R	1.2	○	Wild	●	⊗
DCLDE 2011	6,011	0.7	4.0	○	Wild	○	⊗
SDWD	N/R	N/R	43+	● (293)	Wild (C&R)	○	◐

* Estimated from total vocalization duration and mean whistle duration.

THANK YOU

Questions?

ML on Temporal Graphs · Linguae-ML Seminar · 19/05/2026