EURANOVA

Copenhagen,
June 2022

Industrial challenges of evolving graph neural networks (GNNs)

DEBS 2022

Madalina Ciortan, Head of Data Science @ EURANOVA

EURA NOVA

A DATA COMPANY
AI CHAMPION IN BE

Use data to explore, craft & serve the vision of leading companies

EURANOVA

Serve
your business ambition from POC to production.

data consulting

Craft
solutions to speed up your process.

solutions incubator

Explore
future technology to address today's challenges.

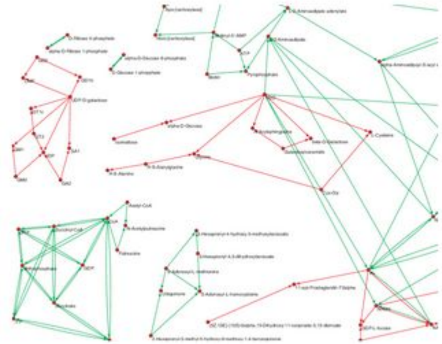
research centre

HIGH LEVEL CONSULTING SERVICES

Context

Large graphs are nowadays ubiquitous

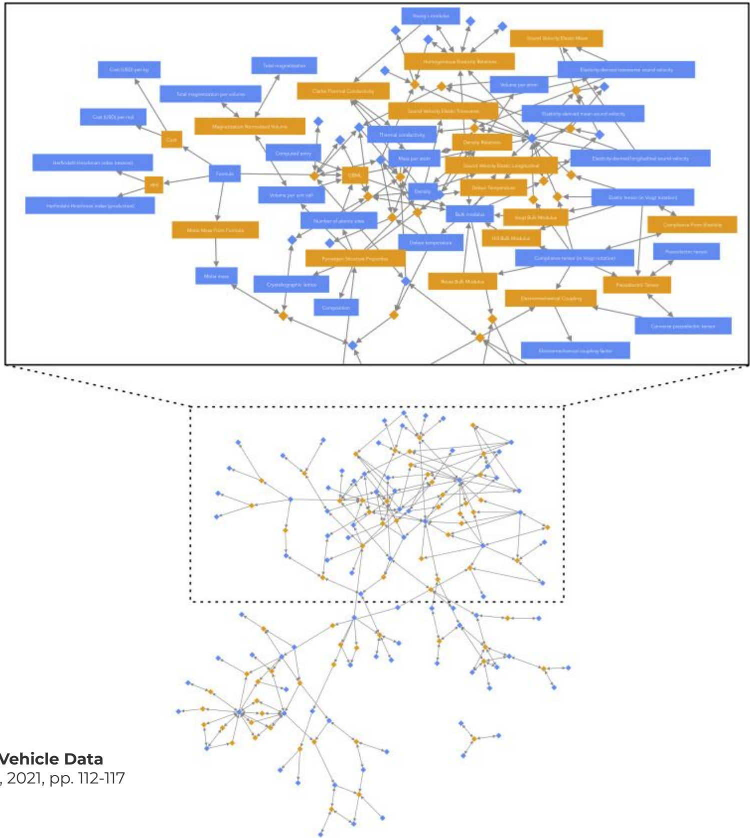
Where graphs get used



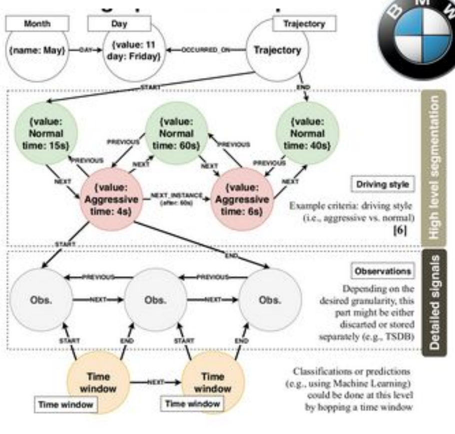
Pharma
Metabolic pathways & genetic regulation

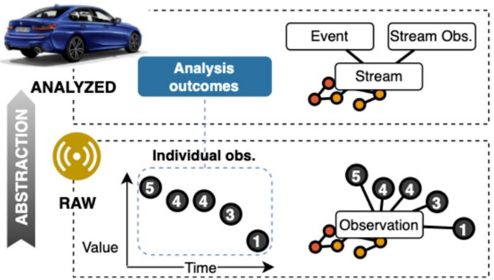
Material Science
Knowledge graph for material science

Mrdjenovich, David & Horton, Matthew & Montoya, Joseph & Legaspi, Christian & Dwaraknath, Shyam & Tshitoyan, Vahe & Jain, Anubhav & Persson, Kristin. (2020). **propnet: A Knowledge Graph for Materials Science.** Matter. 10.1016/j.matt.2019.11.013.



Automotive
Connected car graph profile





D. Alvarez-Coello, D. Wilms, A. Bekan and J. M. Gómez, **"Generic Semantization of Vehicle Data Streams,"** *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 112-117

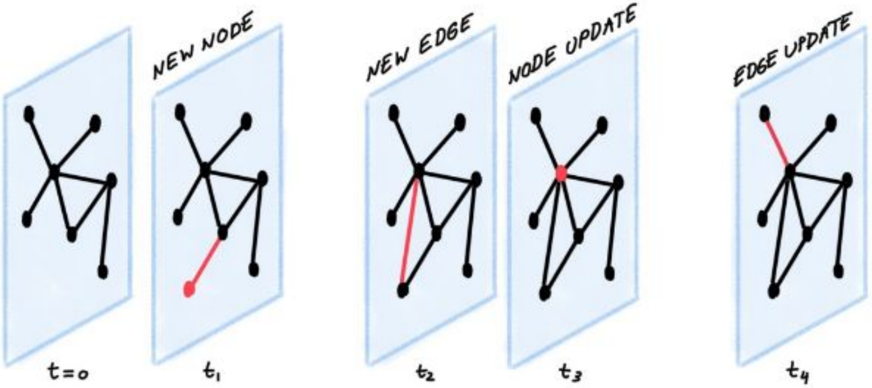
What about new data?

Temporal/Evolving graph networks

Plasticity/Stability dilemma:

- **New** important **patterns** may **appear**
- Some **historical patterns** remain **relevant**

Temporal/ Evolving/Dynamic graphs



https://blog.twitter.com/engineering/en_us/topics/insights/2021/temporal-graph-networks

Challenges

- Constant updates → **detect new patterns** while keeping previous knowledge
- Large graphs → **memory efficient** models are needed
- We need **accurate**, **fast** and **trustworthy** models

→ We research and contribute to:

- Model **explainability**
- **Continual** learning

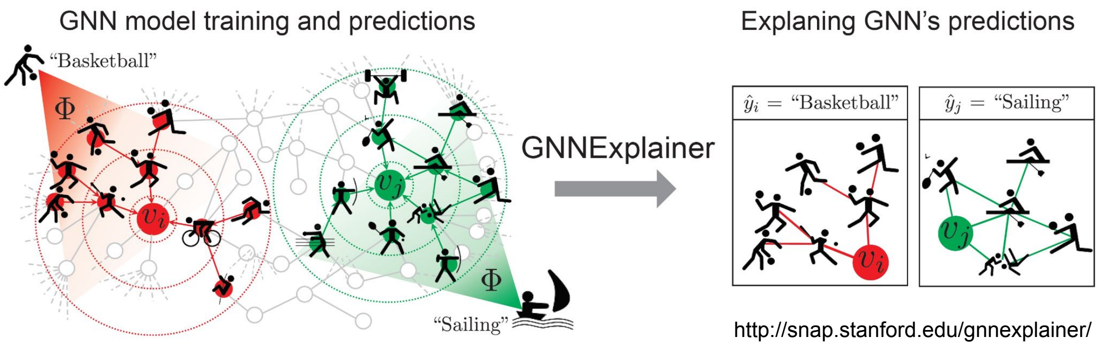
Explainability

What is GNN model explainability?

Identifying the:

- **Nodes/ node features**
- **Edges/ edge features**

used by the GNN model to make predictions



http://snap.stanford.edu/gnnexplainer/

→ create an **explanatory subgraph**

- **Sparsity**: minimum set of selected features (nodes/edges/features)
- **Stability**: similar explanations for similar inputs
- **Fidelity** of predictions: minimum difference between whole graph prediction and the subgraph prediction

How can we explain GNNs?

- Generalize explainability methods designed for traditional NN
- Graph topology introduces **additional challenges**
- Explainability methods for **static** graphs can be applied to **evolving** graphs

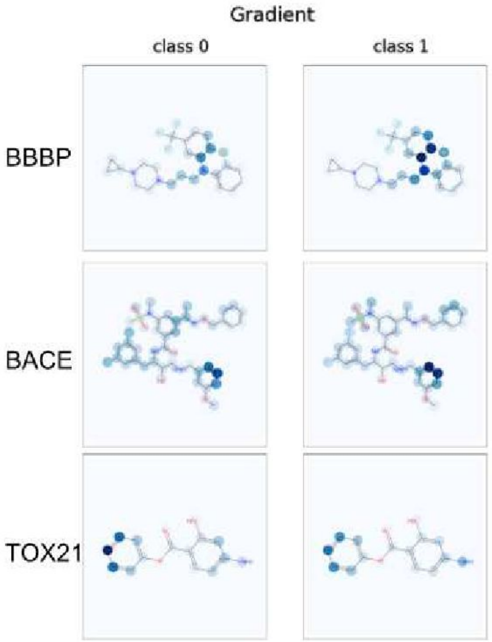
1. Gradient-based	**2. Decomposition**	**3. Perturbation-based**	**4. Surrogate models**

Gradient based methods

1. Gradient-based | 2. Decomposition | 3. Perturbation-based | 4. Surrogate models

- Larger **gradients** or **hidden feature-map** values indicate higher input feature importance

- *Examples*: CAM, Grad-CAM, SA, Guided BP

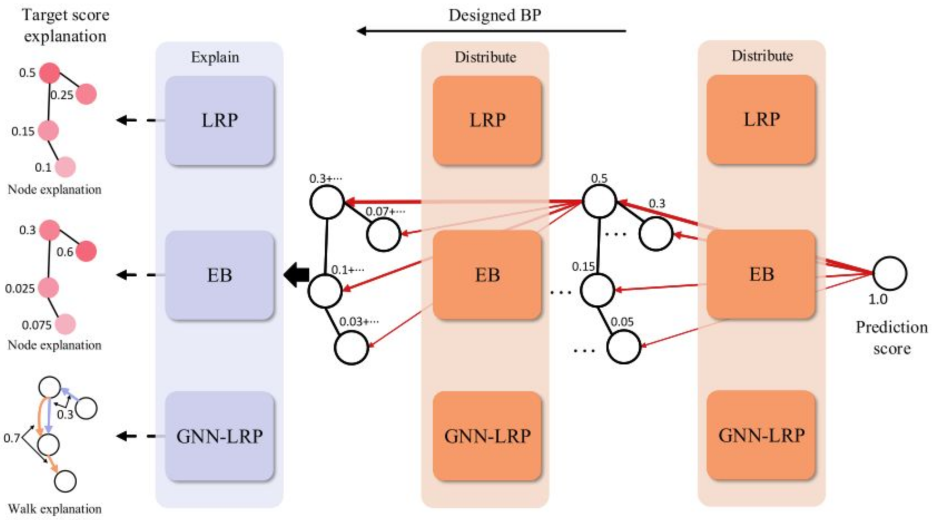


Decomposition methods

1. Gradient-based	**2. Decomposition**	3. Perturbation-based	4. Surrogate models

- The prediction score is decomposed layer-by-layer until the input into sums of importance scores

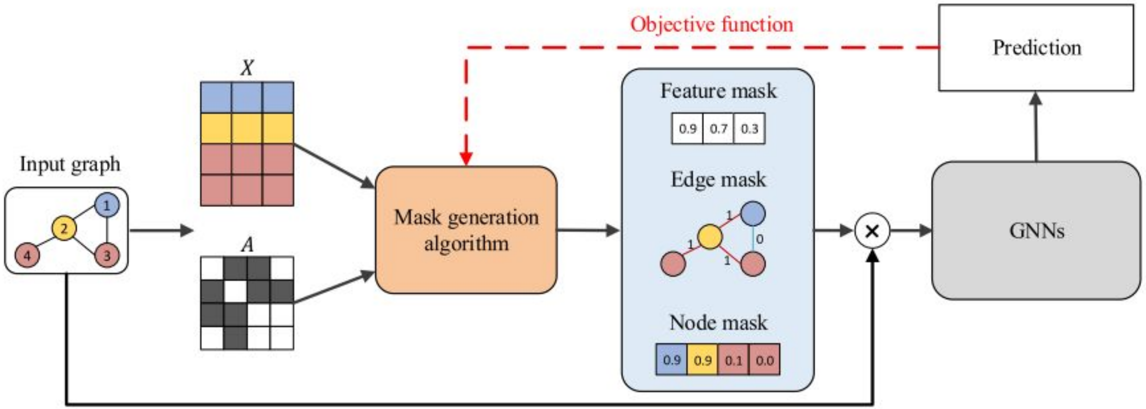
- *Examples*: LRP, GNN-LRP, ExcitationBP



Perturbation based methods

1. Gradient-based	2. Decomposition	**3. Perturbation-based**	4. Surrogate models

- A set of **masks (node/ edge)** is learned by input perturbations and identifies the important input features
- *Examples*: GNNExplainer, PGExplainer, ZORRO, GraphMask, SubgraphX
- Methods based on **RL** (e.g. RGExplainer) are currently studied at ENX



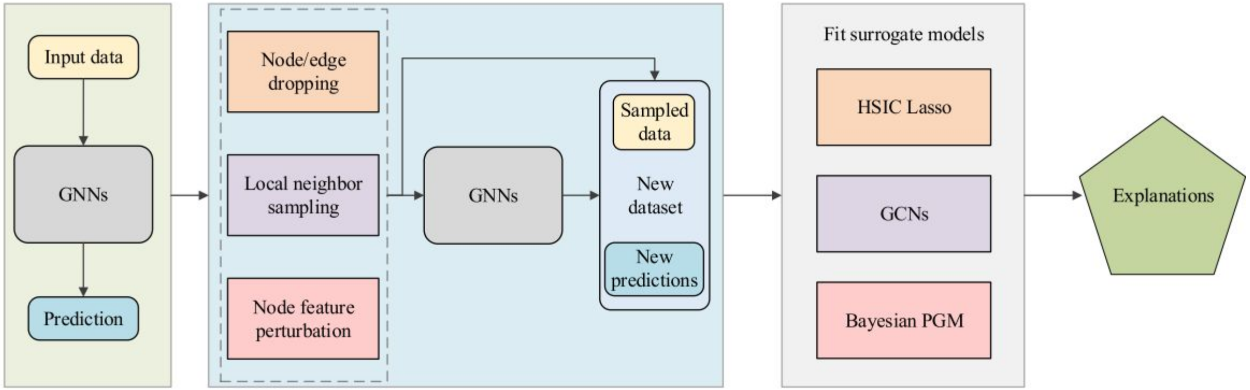
[1]. Explainability in GNNS: A Taxonomic Survey, Hao Yuan et. al

Surrogate methods

1. Gradient-based	2. Decomposition	3. Perturbation-based	**4. Surrogate models**

- **Simple** and **interpretable surrogate** model to approximate the complex predictions of GNNs

- *Examples*: GraphLime, RelEx, PGMExplainer



Which method should I use?

- Few surveys [1-3]
- Lack of experimental exercises
- No clear guidelines and performance evaluation

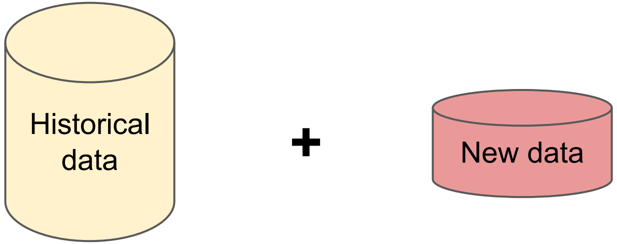
Method	Type	Learning	Task	Target	Black-box	Flow	Design
SA [52], [53]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
Guided BP [52]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
CAM [53]	Instance-level	✗	GC	N	✗	Backward	✗
Grad-CAM [53]	Instance-level	✗	GC	N	✗	Backward	✗
GNNExplainer [44]	Instance-level	✓	GC/NC	E/NF	✓	Forward	✓
PGExplainer [45]	Instance-level	✓	GC/NC	E	✗	Forward	✓
GraphMask [55]	Instance-level	✓	GC/NC	E	✗	Forward	✓
ZORRO [54]	Instance-level	✗	GC/NC	N/NF	✓	Forward	✓
Causal Screening [56]	Instance-level	✗	GC/NC	E	✓	Forward	✓
SubgraphX [46]	Instance-level	✓	GC/NC	Subgraph	✓	Forward	✓
LRP [52], [57]	Instance-level	✗	GC/NC	N	✗	Backward	✗
Excitation BP [53]	Instance-level	✗	GC/NC	N	✗	Backward	✗
GNN-LRP [58]	Instance-level	✗	GC/NC	Walk	✗	Backward	✓
GraphLime [59]	Instance-level	✓	NC	NF	✓	Forward	✗
RelEx [60]	Instance-level	✓	NC	N/E	✓	Forward	✓
PGM-Explainer [61]	Instance-level	✓	GC/NC	N	✓	Forward	✓
XGNN [43]	Model-level	✓	GC	Subgraph	✓	Forward	✓

[1] Hao Yuan et al . (2021) **"Explainability in Graph Neural Networks: A Taxonomic Survey",** https://arxiv.org/abs/2012.15445
[2] Li, Peibo & Yang, Yixing & Pagnucco, Maurice & Song, Yang. (2022). "**Explainability in Graph Neural Networks: An Experimental Survey**"
[3] Dai, Enyan and Zhao (2022). "**A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability**"

Continual learning

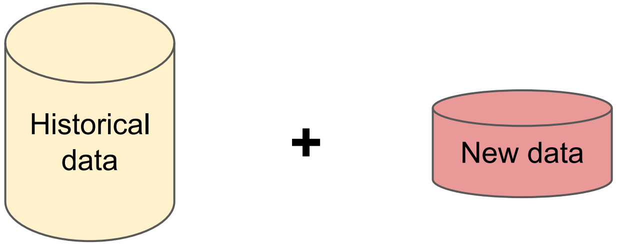
Strategies to deal with new tasks/data

Approach	Description
Joint training	Train arbitrarily on **old + new** data → *High computational cost*
Online learning	Train on **new** data → *Catastrophic forgetting*
Continual learning	Remember **existing** information and capture **new** patterns

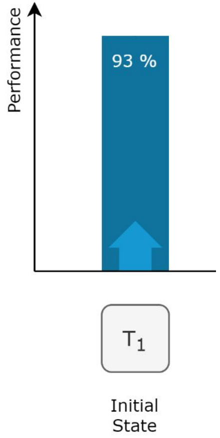


Strategies to deal with new tasks/data

Approach	Description
Joint training	Train arbitrarily on **old + new** data → *High computational cost*
Online learning	Train on **new** data → *Catastrophic forgetting*
Continual learning	Remember **existing** information and capture **new** patterns

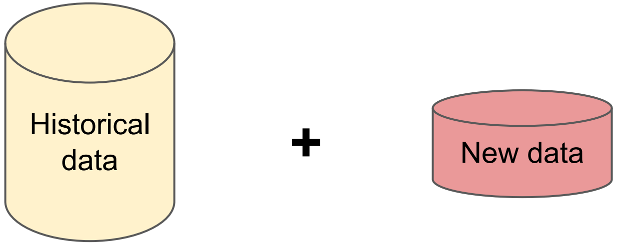


Catastrophic forgetting

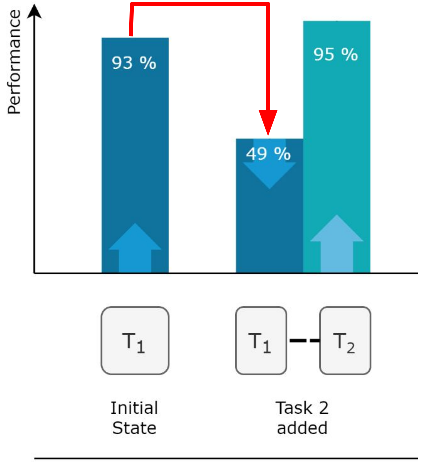


Strategies to deal with new tasks/data

Approach	Description
Joint training	Train arbitrarily on **old + new** data → *High computational cost*
Online learning	Train on **new** data → *Catastrophic forgetting*
Continual learning	Remember **existing** information and capture **new** patterns

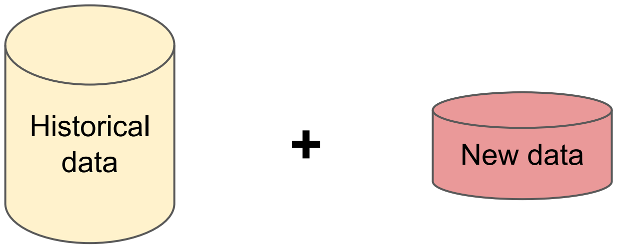


Catastrophic forgetting

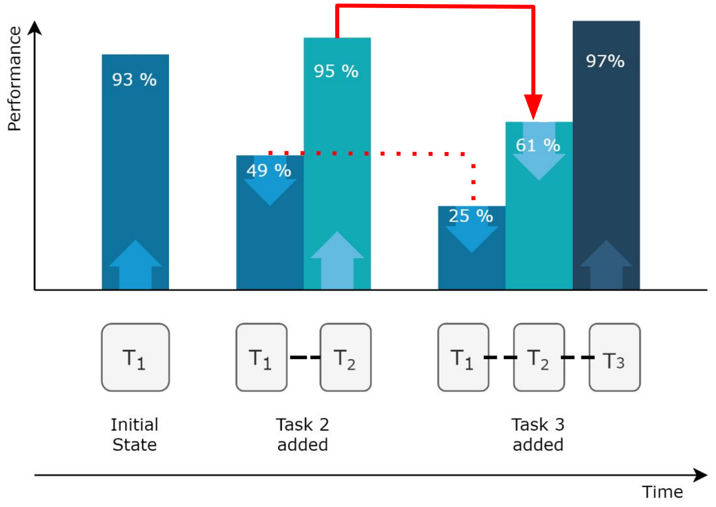


Strategies to deal with new tasks/data

Approach	Description
Joint training	Train arbitrarily on **old + new** data → *High computational cost*
Online learning	Train on **new** data → *Catastrophic forgetting*
Continual learning	Remember **existing** information and capture **new** patterns



Catastrophic forgetting



Continual learning

- Terminology: **lifelong** learning = **continual** learning = **incremental** learning = **never-ending** learning
- The objective is to learn a task T(t) with graph data G(t) while maintaining the performance and avoiding the **catastrophic forgetting** for T(1), T(2), … ,T(t-1)
- We can consider **new evolving data** as a new task (e.g. Traffic Stream model)

$$G(t) = G(t-1) + \Delta G(t)$$

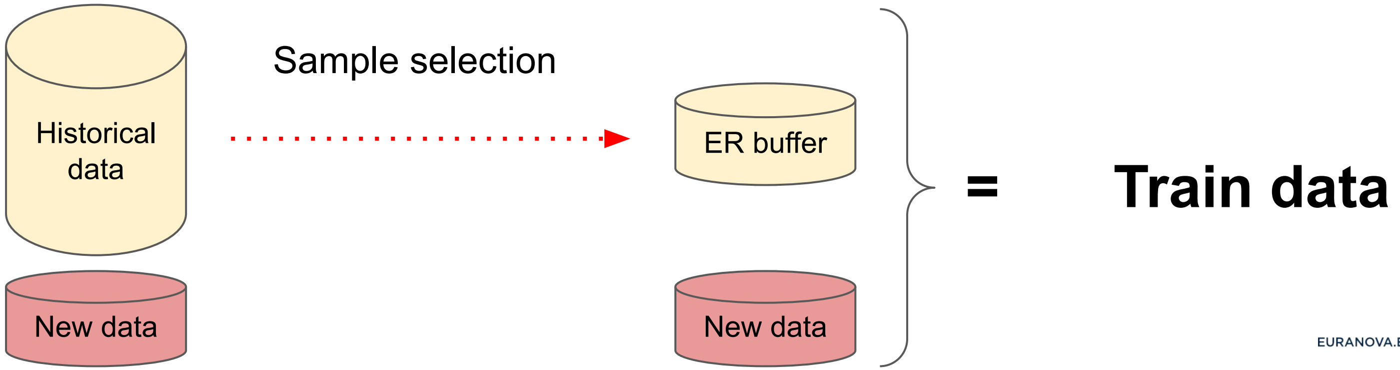
1. Rehearsal Approach	2. Regularization Approach	3. Architectural Approach

Rehearsal approaches

1. Rehearsal Approach | **2. Regularization Approach** | **3. Architectural Approach**

- Keep a buffer of representative samples of each past task (**Experience Replay**) to be combined with new data
- **Sample selection** method

- *Examples*: ER-GNN, Open-world Node Classification

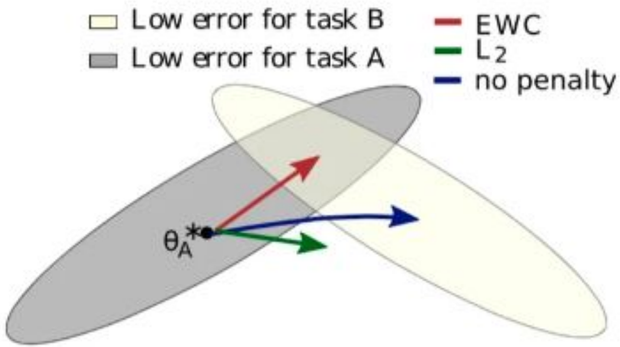


Regularization approaches

1. Rehearsal Approach	**2. Regularization Approach**	3. Architectural Approach

- Design dedicated **loss functions**
- Most popular example is EWC (Elastic Weight Consolidation), which penalizes the updates to the weights important to previous tasks

- *Examples*: Continual GNN



Architectural approaches

1. Rehearsal Approach	2. Regularization Approach	**3. Architectural Approach**

- Change the **architecture** (layers/activation functions) of the GNN to address a new task and prevent forgetting previous ones
- A baseline example is adding classification heads/layers for each new task (e.g. Progressive NNs)
- Existing methods (on CNNs) assume the samples are independent, which is not hold in graphs

- *Example*: Feature graphs decorrelate the neighborhood of nodes

Conclusion

1 **No consensus**
on the best method for explainability /continual learning

2 **Not enough guidance**
to make a contextual educated choice based on the advantages and weaknesses of existing methods

3 **We are creating a survey/library**
based on extensive experimentation to address these drawbacks

4 **Apply existing methods**
and our contributions to real-world streaming data (e.g. automotive)

Questions?

Backup slides

Lifelong graph learning

Traffic forecasting Paper

Scenarios of using GNNs on temporal data

1. Use a **pretrained** GNN in an inductively on new data → the **performance degrades** gradually
2. **Retrain** the **whole network** regularly when new data arrives → **high** computational **complexity**
3. **Online** learning trains **only on the new data** (ΔG)→ can lead to **catastrophic forgetting**
4. **Continual** learning→ capture new patterns incrementally and consolidate existing information

$$G(t) = G(t-1) + \Delta G(t)$$

Graph Lifelong learning: survey

- Terminology: **lifelong** learning = **continual** learning = **incremental** learning = **never-ending** learning
- The objective is to learn a task T(t) with graph data G(t) while maintaining the performance and avoiding the catastrophic forgetting for T(1), T(2), … ,T(t-1)

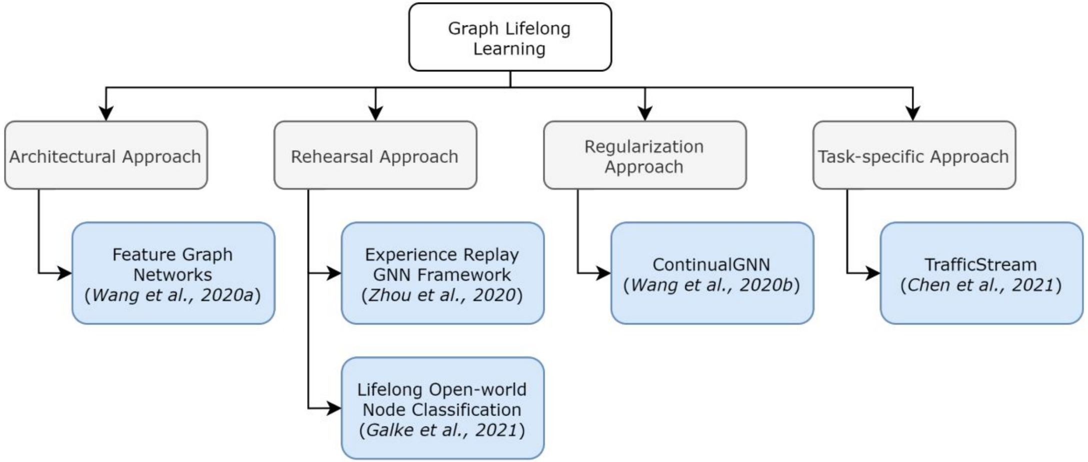


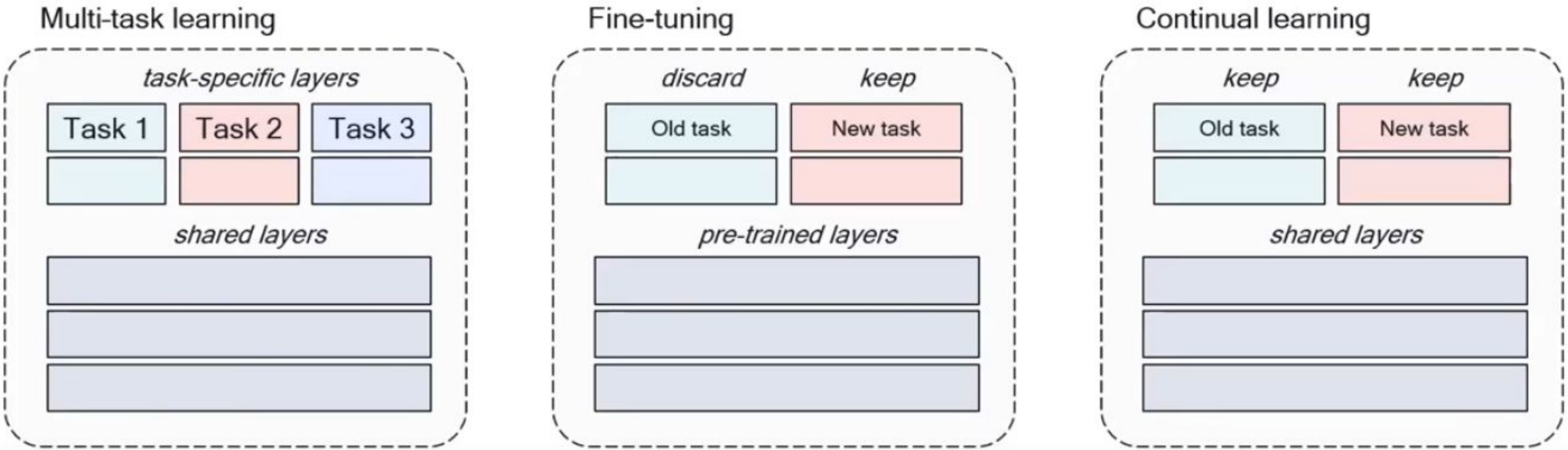
Figure 2: Graph Lifelong Learning Categorization.

Conclusion

- No consensus on the best method for explainability /continual learning
- Not enough guidance to make a contextual educated choice based on the advantages and weaknesses of existing methods

- We are creating a survey/library based on extensive experimentation to address these drawbacks
- Apply existing methods and our contributions to real-world streaming data (e.g. automotive)

Continual learning



Context

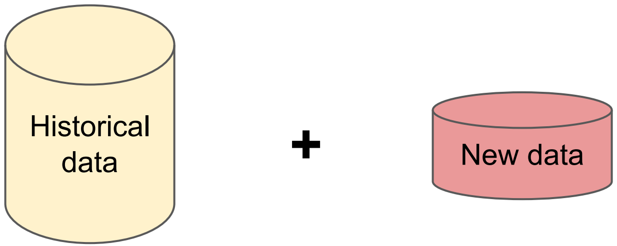
- most contributions on explainability study **static** graphs
- only a few dedicated methods for temporal graphs
  - xRTE, RetaGNN, TLogic
  - based on attention
  - don't evaluate the explainability per se
-

Context

- temporal graph data (e.g. traffic flow) consists of intricate spatial-temporal correlations which are ignored by static models
- the objective is to mine new patterns while consolidating historical knowledge (avoiding catastrophic forgetting)
- when a new task is similar to a previous one, improve the existing model; when a new task is different, transfer knowledge to address it and thus be able to handle new undefined tasks

Strategies to deal with new tasks/data

Approach	Description
Joint training	Train arbitrarily on **old + new** data → *High computational cost*
Online learning	Train on **new** data → *Catastrophic forgetting*
Continual learning	Consolidate **existing** information and capture **new** patterns



Catastrophic forgetting