# Explainability and Applicability of Graph Neural Networks

Josephine Thomas, Silvia Beddar-Wiesing, Clara Holzhüter, Alice Moallemy-Oureh

October 23, 2023

Intelligent Embedded Systems
UNIKASSEL VERSITÄT
Fraunhofer IEE
Bundesministerium für Bildung und Forschung

# Content

# The Team

Josephine Thomas

Silvia Beddar-Wiesing

Alice Moallemy-Oureh

Clara Holzhüter

Bernhard Sick

Christoph Scholz

Björn-André Schröder

Laura Ritter

We are looking for more student assistants!

# Schedule

Preliminary Schedule

| Wednesday | | Thursday | | Friday | |
|---|---|---|---|---|---|
| 10:00-11:10 | TBD<br>GAIN | 10:00-10:45 | Explaining Identity-aware Graph Classifiers through the Language of Motifs<br>Alan Perotti | 10:00-12:00 | Hands-on Tutorial on Explaing GNNs<br>Dominik Köhler |
| 11:15-12:00 | How can we use random walks in deep learning on graphs and why do we care?<br>Martin Ritzert | 10:50-11:35 | The most important unsolved problem in graph representation learning<br>Petar Velickovic* | Afterwards | Lunch |
| 12:00-13:00 | Lunch | 11:35-11:50 | Coffee | | |
| 13:00-13:45 | Graph Neural Networks for Power Systems Operation<br>Balthazar Donon | 11:50-12:35 | Deep Learning on Real-World Graphs<br>Emanuele Rossi | | |
| 13:50-14:35 | Weisfeiler and Leman go Neural: Expressivity and Generalization Abilities of Graph Neural Networks<br>Christopher Morris | 12:35-15:00 | Break | | |
| 14:35-15:00 | Coffee | 15:00-Open end | Social Event | | |
| 15:00-15:45 | Network Optimization with GNNs and Deep Reinforcement Learning<br>Paul Almasan | | | | |
| 15:50-16:35 | Approximately equivariant graph networks<br>Soledad Villar* | | | | |
| 17:00-17:45 | Reliable Graph Machine Learning<br>Simon Geisler | | | | |

*Talks will be online talks

# Social event

Source: `https://www.ro80club.org/`

- Meeting point: Here or tram stop 'Wilhelmshöhe Park' in front of the information!
- Time: 15.00 here or about 15.45 at 'Wilhelmshöhe Park'.
- Dinner: 19.00 at restaurant Lichtenhainer (Elfbuchenstraße 4 34119 Kassel)

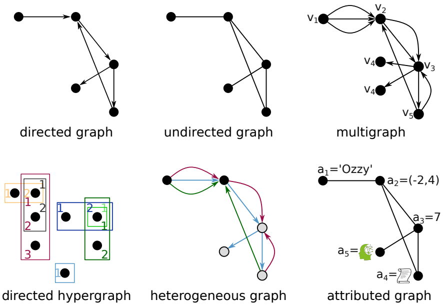
# GAIN Research: Past, current and future work

Josephine Thomas

# Past work

## Graph Neural Networks Designed for Different Graph Types: A Survey

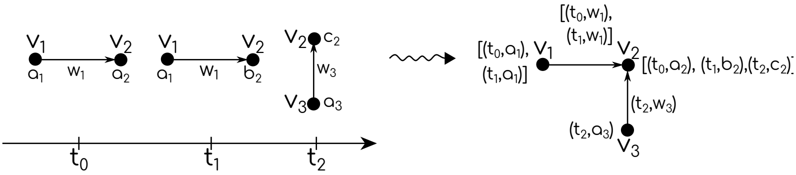
published at TMLR




Josephine Thomas*, Alice Moallemy-Oureh*, Silvia Beddar-Wiesing*, Clara Holzhüter: *Graph Neural Networks Designed for Different Graph Types: A Survey*, Transactions on Machine Learning Research, 2023, https://openreview.net/forum?id=h4BYtZ79uy

## A Note on the Modeling Power of Different Graph Types

preprint available




Josephine M. Thomas, Silvia Beddar-Wiesing, Alice Moallemy-Oureh, Rüdiger Nather: *A Note on the Modeling Power of Different Graph Types*, https://arxiv.org/abs/2109.10708

# Past work

Weisfeiler–Lehman goes Dynamic: An Analysis of the Expressive Power of Graph Neural Networks for Attributed and Dynamic Graphs
under review at Neural Networks

- Which graphs/nodes can a GNN distinguish?
- Which functions can a GNN approximate?

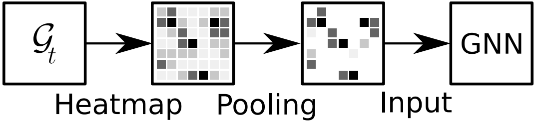
  $\rightarrow$ Extension of the work of D'Inverno et. al (2021) and Azizian et. al (2020) from static node-attributed graphs to dynamic and fully attributed graphs

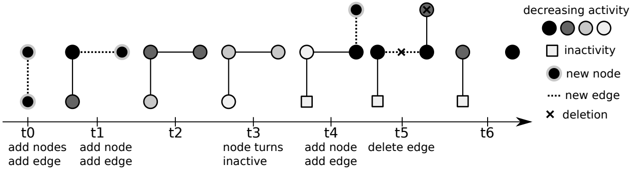
Beddar-Wiesing*, D'Inverno*, Graziani*, Lachi*, Moallemy-Oureh*, Scarselli, Thomas: *Weisfeiler–Lehman goes Dynamic: An Analysis of the Expressive Power of Graph Neural Networks for Attributed and Dynamic Graphs*, https://arxiv.org/abs/2210.03990

# Past work

## Using local activity encoding for dynamic graph pooling in stuctural-dynamic graphs: student research abstract

published at ACM



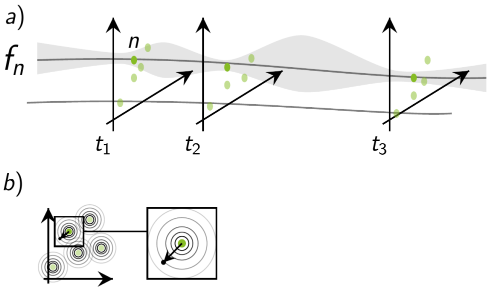




Silvia Beddar-Wiesing: *Using local activity encoding for dynamic graph pooling in stuctural-dynamic graphs: student research abstract*, SAC '22: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing

# Past work

## Continuous-Time Generative GNN for Attributed Dynamic Graphs: student research abstract

published at ACM

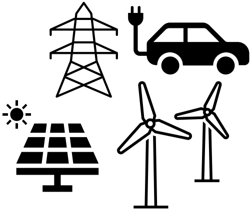


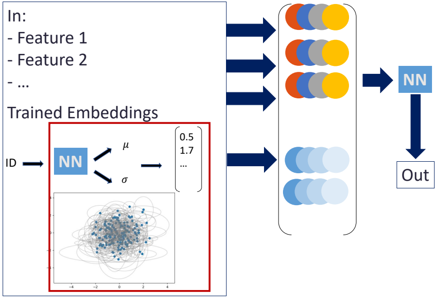

Alice Moallemy-Oureh: *Continuous-time generative graph neural network for attributed dynamic graphs: student research abstract*, SAC '22: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, https://doi.org/10.1145/3477314.3508018

# Past work

## Power flow forecasts at transmission grid nodes using Graph Neural Networks

published at Energy and AI






Dominik Beinert*, Clara Holzhüter*, Josephine M. Thomas, Stephan Vogt: *Power flow forecasts at transmission grid nodes using Graph Neural Networks*, Energy and AI 2023, `https://www.sciencedirect.com/science/article/pii/S2666546823000344`

## Current and future work

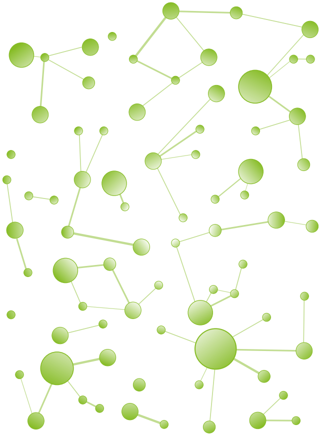
- Explainability of our algorithms
- Implementation of FDGNN
- Implementation of algorithms for structural dynamic and attribute dynamic graphs
- Combining Reinforcement Learning with Graph Learning for use-cases on the power grid

# Current and future work

- Explainability of our algorithms
- Implementation of FDGNN
- Implementation of algorithms for structural dynamic and attribute dynamic graphs
- Combining Reinforcement Learning with Graph Learning for use-cases on the power grid

- Explainability of our algorithms
- Implementation of FDGNN
- Implementation of algorithms for structural dynamic and attribute dynamic graphs
- Combining Reinforcement Learning with Graph Learning for use-cases on the power grid

- Explainability of our algorithms
- Implementation of FDGNN
- Implementation of algorithms for structural dynamic and attribute dynamic graphs
- Combining Reinforcement Learning with Graph Learning for use-cases on the power grid



# Explainability in Graph Neural Networks

Josephine Thomas

# Explainability: Motivation


Source: https://carpentries-incubator.github.io/data-science-ai-senior-researchers/05-Problems-with-AI/index.html

We believe, the algorithm learned to classify wolfes and huskys with 80% accuracy...

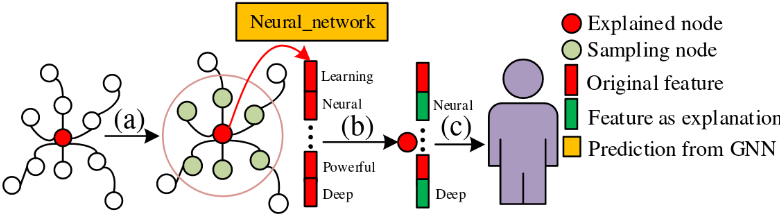
# Explainability: Motivation

(a) Husky classified as wolf (b) Explanation

LIME, Ribeiro et al. 2016

..but it actually learned to recognize snow/bright background.

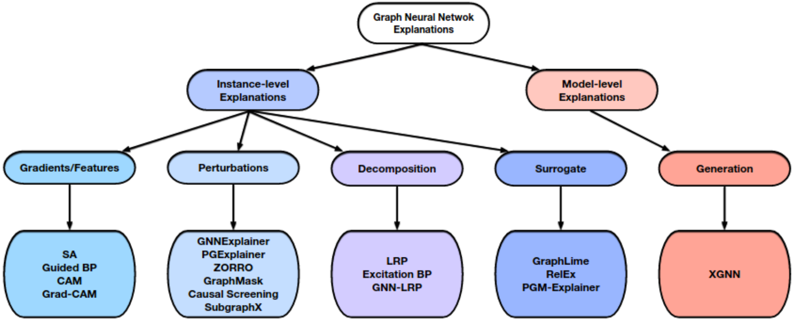
# Explainability for GNNs, an example: GraphLIME




GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks, Huang et al., 2023

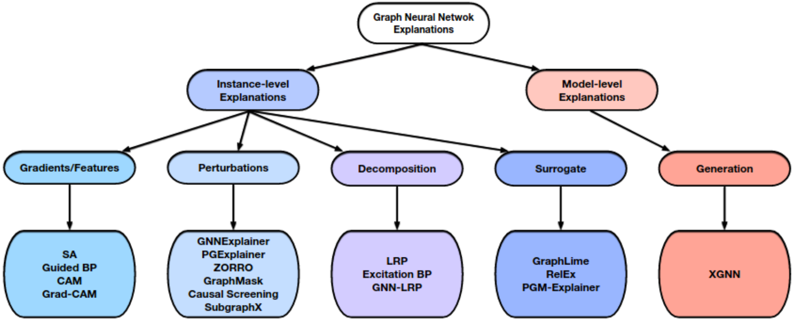
For graphs, the most representative features of a nodes neighbors can be selected to serve as an explanation for the classification result of that node.

# Explainability: Types of Explainability for GNNs




Explainability in Graph Neural Networks: A Taxonomic Survey, Yuan et al. 2022, IEEE Transactions on Pattern Analysis and Machine Intelligence

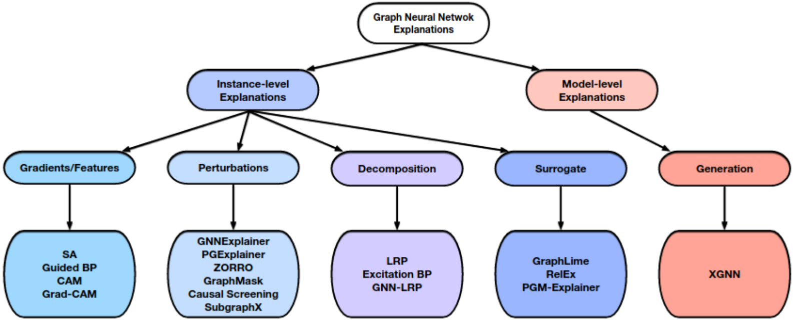
# Explainability: Types of Explainability for GNNs




Explainability in Graph Neural Networks: A Taxonomic Survey, Yuan et al. 2022, IEEE Transactions on Pattern Analysis and Machine Intelligence

- Model-specific or model-agnostic?

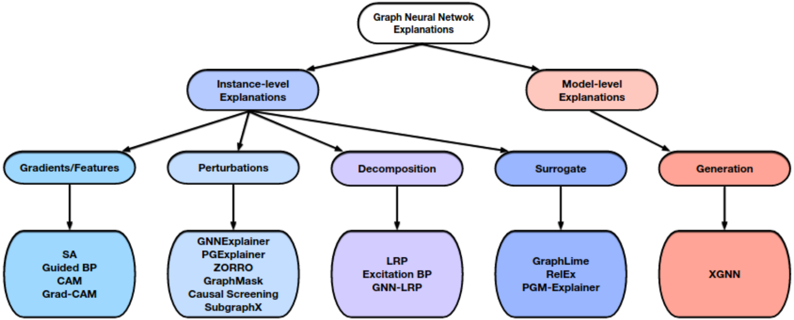
# Explainability: Types of Explainability for GNNs




Explainability in Graph Neural Networks: A Taxonomic Survey, Yuan et al. 2022, IEEE Transactions on Pattern Analysis and Machine Intelligence

- Model-specific or model-agnostic?
- Local or global?

# Explainability: Types of Explainability for GNNs




Explainability in Graph Neural Networks: A Taxonomic Survey, Yuan et al. 2022, IEEE Transactions on Pattern Analysis and Machine Intelligence

- Model-specific or model-agnostic?
- Local or global?
- Post-hoc or inherent?

**Use case power grid:**

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])

[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])
- Need for speed

[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])
- Need for speed

**Dynamic graphs:**

[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])
- Need for speed

**Dynamic graphs:**

- A lot of explainability methods for GNNs on static graph so far

[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])
- Need for speed

**Dynamic graphs:**

- A lot of explainability methods for GNNs on static graph so far

**How good is the answer?**

[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

# Explainability: What we need to consider to explain our models

**Use case power grid:**

- Extreme need for safety
  - Local
  - Inherent (no need to explain the explainers..[1])
- Need for speed

**Dynamic graphs:**

- A lot of explainability methods for GNNs on static graph so far

**How good is the answer?**

- faithfulness, sparsity, correctness and plausibility [2][3] ...

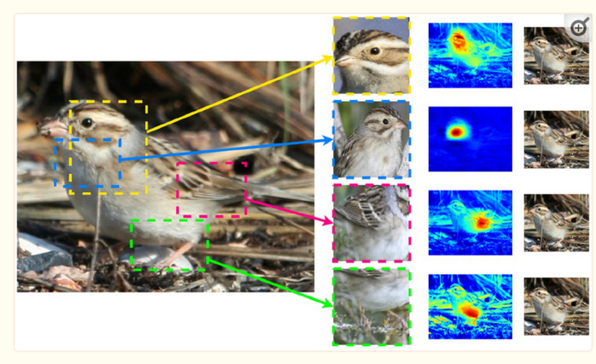
[1] Explaining the Explainers in Graph Neural Networks: a Comparative Study A. Longa et al., 2022

[2] BAGEL, Rathee et al. 2022, https://arxiv.org/pdf/2206.13983.pdf

[3] GraphXAI, Agarwal et al. 2023, https://www.nature.com/articles/s41597-023-01974-x

# Explainability: Prototype-based explanations

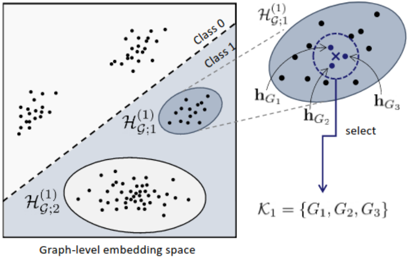
This bird is a clay-colored sparrow, because it has the prototypical wing/eyes...



Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead, Cynthia Rudin, 2019, Nat Mach Intell.

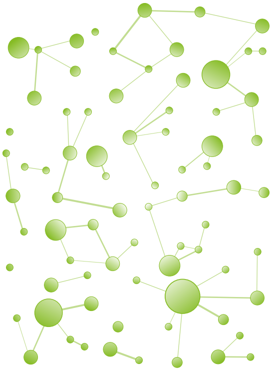
# Explainability: Prototype-based explanations

Discovering human-interpretable prototype graphs [1] is a similar method for graphs.



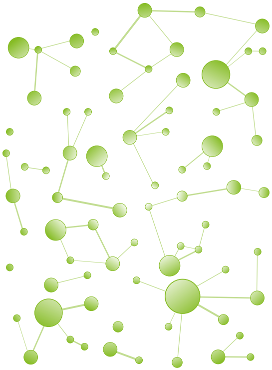

PAGE: Prototype-Based Model-Level, Explanations for Graph Neural Networks, Shin et al., 2022, `https://arxiv.org/pdf/2210.17159.pdf`

Do you agree this is a promising method to explain our algorithms?



# Power flow forecasts at transmission grid nodes using Graph Neural Networks

Clara Holzhüter



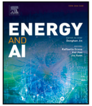
# Power flow forecasts at transmission grid nodes using Graph Neural Networks

Clara Holzhüter

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai




## Power flow forecasts at transmission grid nodes using Graph Neural Networks

Dominik Beinert [a,1], Clara Holzhüter [a,b,*,1], Josephine M. Thomas [b], Stephan Vogt [b]

- power grids are increasingly complex

- power grids are increasingly complex
- Generation: Renewable energies fluctuate a lot

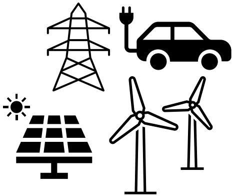
- power grids are increasingly complex
- Generation: Renewable energies fluctuate a lot
- Consumption: more volatile due to electrification

# Power flow forecasts at transmission grid nodes using GNNs

## Introduction

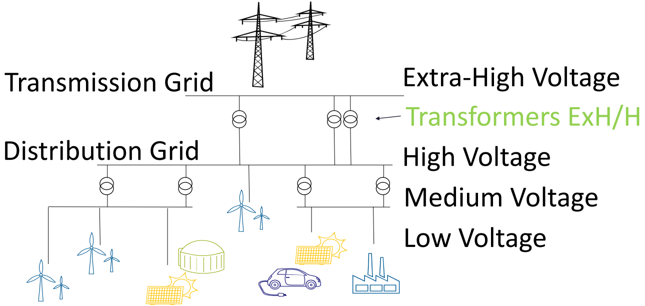
- power grids are increasingly complex
- Generation: Renewable energies fluctuate a lot
- Consumption: more volatile due to electrification

→ Forecasting grid congestion becomes more difficult



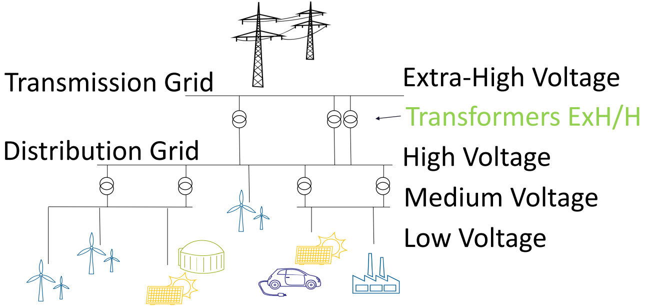
# Power flow forecasts at transmission grid nodes using GNNs

## Use Case: Vertical Power Flow



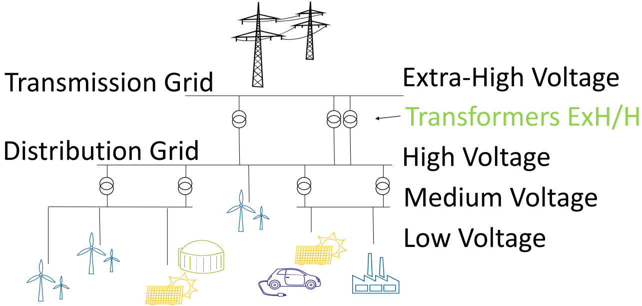
# Power flow forecasts at transmission grid nodes using GNNs
## Use Case: Vertical Power Flow

- Power is generated more decentralized



## Use Case: Vertical Power Flow

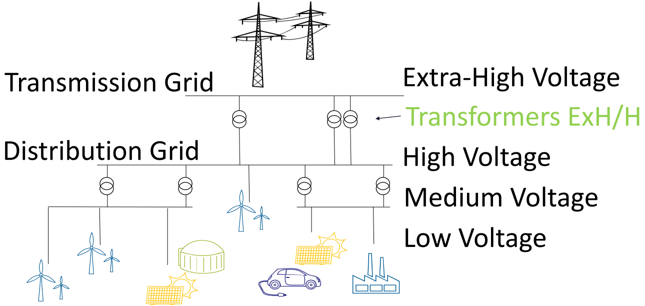
- Power is generated more decentralized
- More power generation in the distribution grid



# Power flow forecasts at transmission grid nodes using GNNs

## Use Case: Vertical Power Flow

- Power is generated more decentralized
- More power generation in the distribution grid



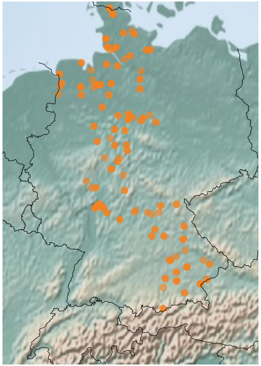

→ Altered power flow complicates grid calculations

## Use Case: Transformers

# Power flow forecasts at transmission grid nodes using GNNs

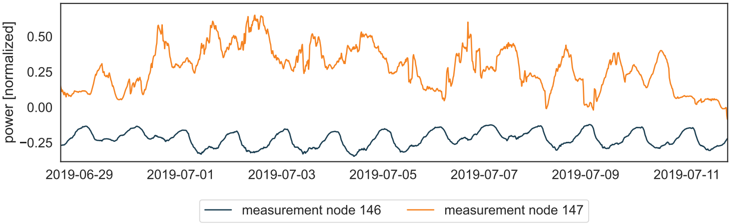
## Use Case: Transformers

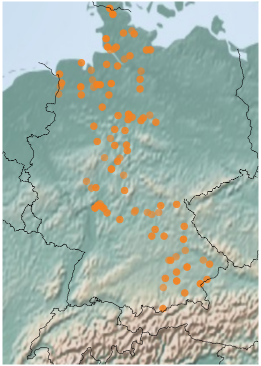
- Locations of transformers influence the power flow patterns through
  - weather
  - Mix of generation
  - consumption pattern
  - ...



## Use Case: Transformers

- Locations of transformers influence the power flow patterns through
  - weather
  - Mix of generation
  - consumption pattern
  - ...



- Power Flows at transformers influence each other

## Use Case: Transformers

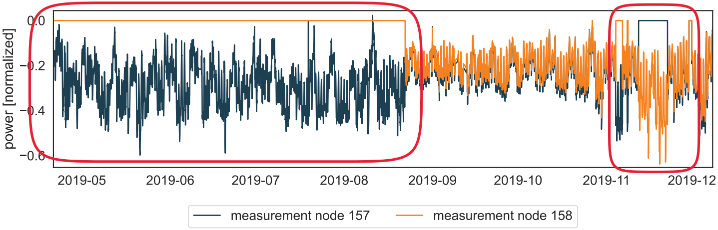
- Power Flows at transformers influence each other
    - Congestion

- Power Flows at transformers influence each other
  - Congestion
  - Grid switching actions

# Power flow forecasts at transmission grid nodes using GNNs

## Use Case: Transformers

- Power Flows at transformers influence each other
    - Congestion
    - Grid switching actions
    - Maintenance



## Implications for the Model

An according prediction model should consider:

## Implications for the Model

An according prediction model should consider:

- Individual characteristics of transformers

An according prediction model should consider:

- Individual characteristics of transformers →Multi-Task

## Implications for the Model

An according prediction model should consider:

- Individual characteristics of transformers $\rightarrow$Multi-Task
- Interactions between transformers

## Implications for the Model

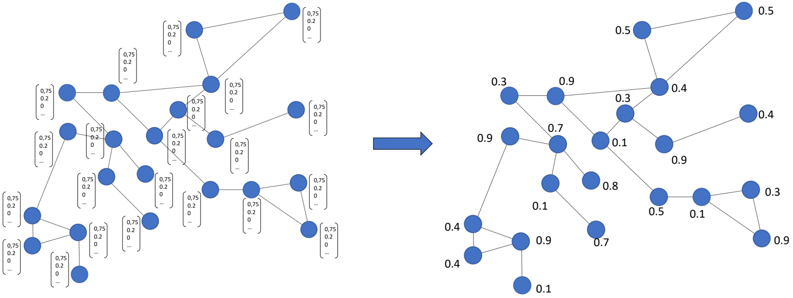
An according prediction model should consider:

- Individual characteristics of transformers $\rightarrow$Multi-Task
- Interactions between transformers $\rightarrow$GNN model

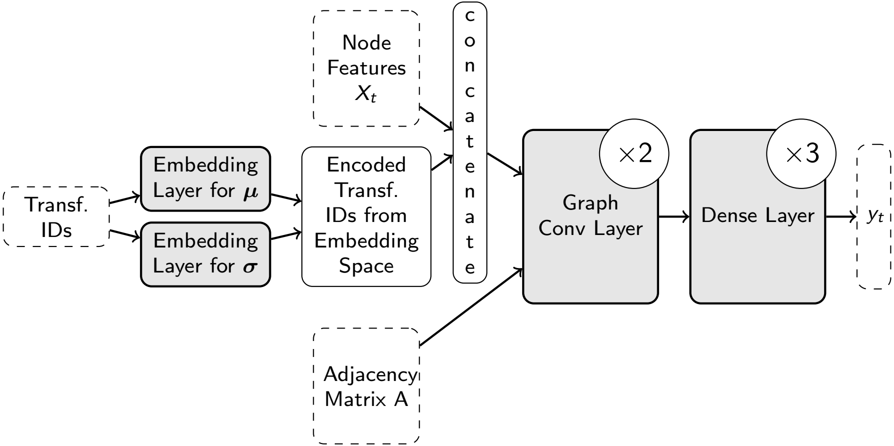
# Power flow forecasts at transmission grid nodes using GNNs

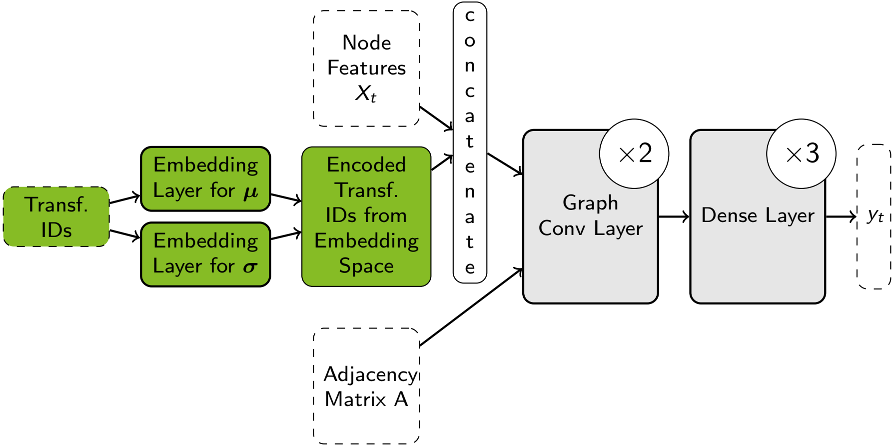
## Problem Setup

- Input: A Set of tranformers and corresponding features
- Output: Power flow at each tranformer



## Our Approach



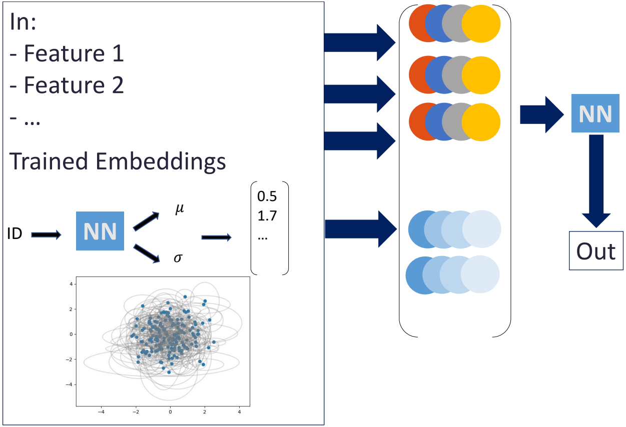

Node Features $X_t$
concatenate
Transf. IDs
Embedding Layer for $\mu$
Embedding Layer for $\sigma$
Encoded Transf. IDs from Embedding Space
Graph Conv Layer
$\times 2$
Dense Layer
$\times 3$
$y_t$
Adjacency Matrix A

## Our Approach: Baysian Multi-Task Embedding

Idea: Solve multiple similar tasks by combining knowledge of all tasks during training while still allow for differences

## Our Approach: Baysian Multi-Task Embedding

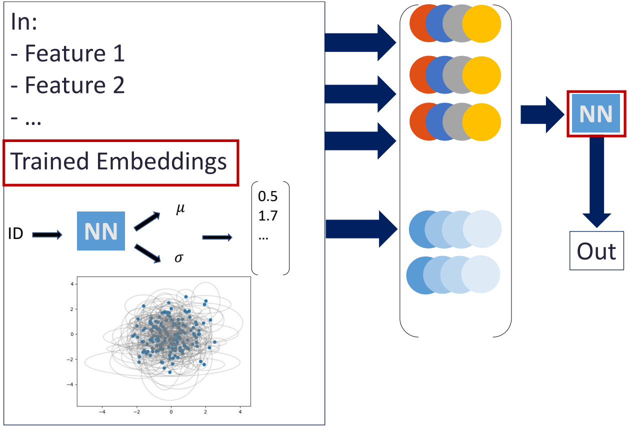
Idea: Solve multiple similar tasks by combining knowledge of all tasks during training while still allow for differences



## Our Approach: Baysian Multi-Task Embedding

Idea: Solve multiple similar tasks by combining knowledge of all tasks during training while still allow for differences

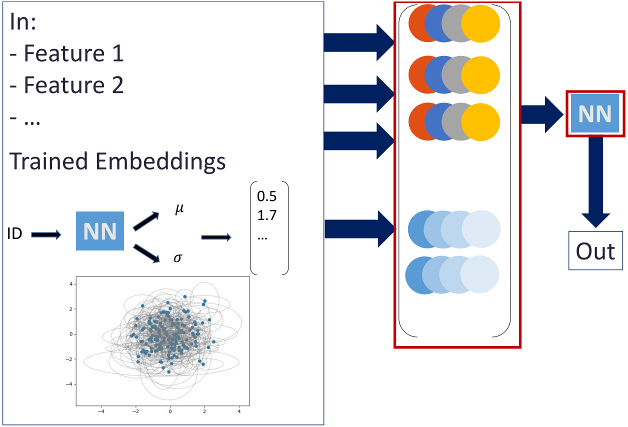
- share weights between all tasks and train individual embedding for each task



## Our Approach: Baysian Multi-Task Embedding

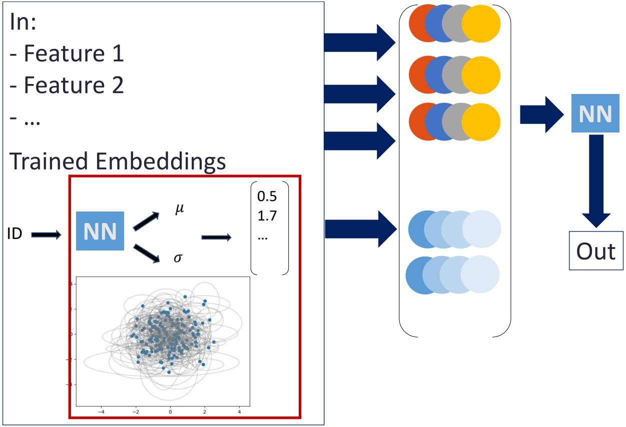
Idea: Solve multiple similar tasks by combining knowledge of all tasks during training while still allow for differences

- share weights between all tasks and train individual embedding for each task
- pass the embedding to the NN in addition to other input variables.



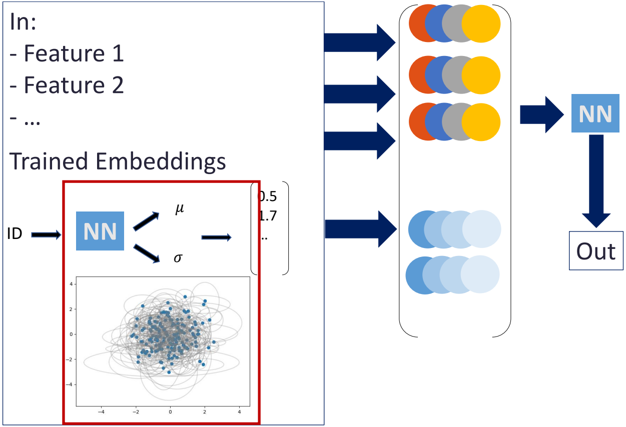
## Our Approach: Baysian Multi-Task Embedding

- Embed the transformers into latent space



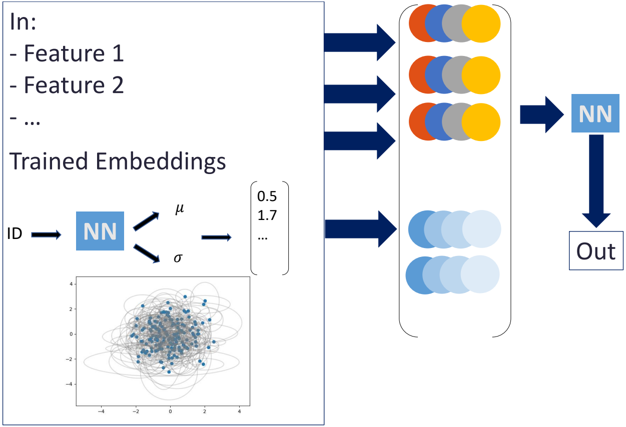
## Our Approach: Baysian Multi-Task Embedding

- Embed the transformers into latent space
- latent representation modelled as multivariate normal distribution

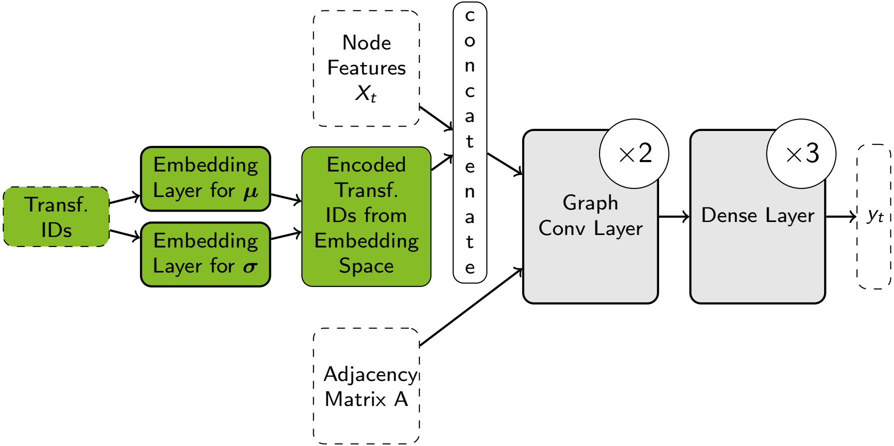


## Our Approach: Baysian Multi-Task Embedding

- Embed the transformers into latent space
- latent representation modelled as multivariate normal distribution
- is trained jointly with overall NN by adding KL-divergence to the loss

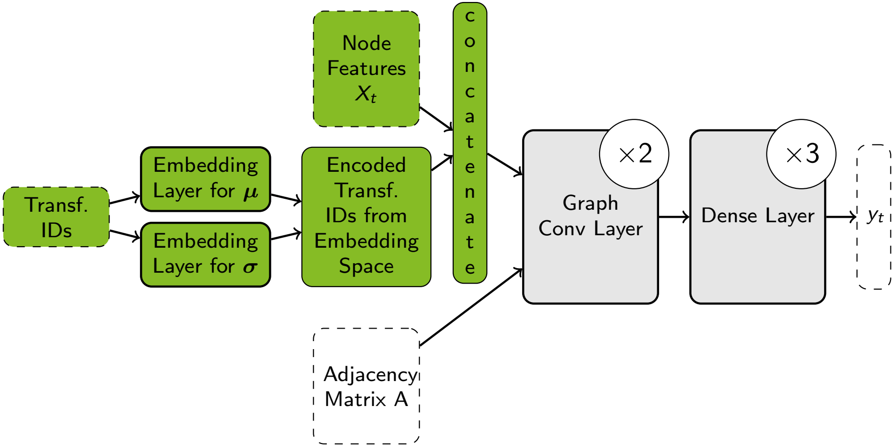


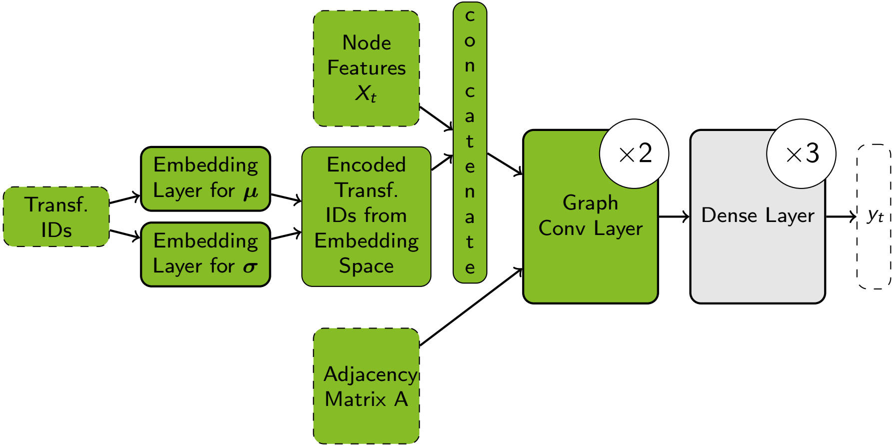
## Our Approach



# Power flow forecasts at transmission grid nodes using GNNs

## Our Approach




Node Features $X_t$
concatenate
Transf. IDs
Embedding Layer for $\mu$
Embedding Layer for $\sigma$
Encoded Transf. IDs from Embedding Space
Graph Conv Layer
$\times 2$
Dense Layer
$\times 3$
$y_t$
Adjacency Matrix A

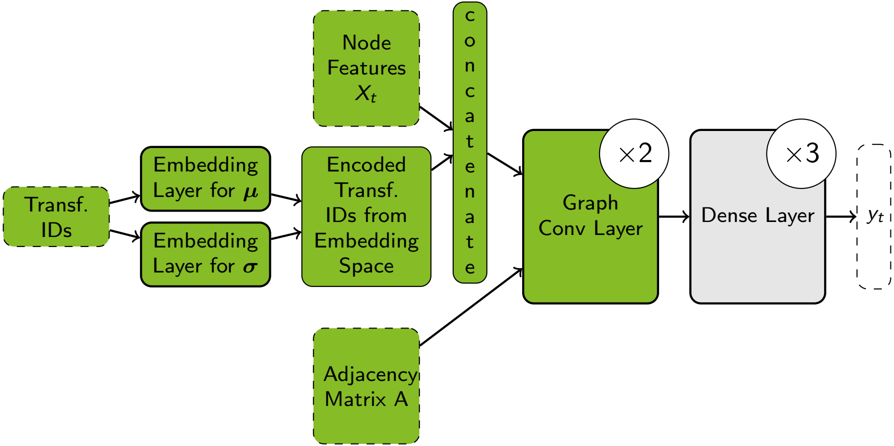
## Our Approach: GNN Model

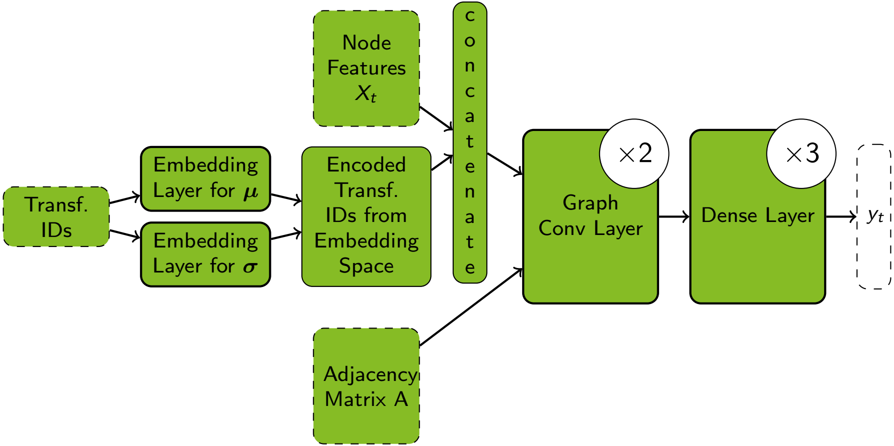
Architecture

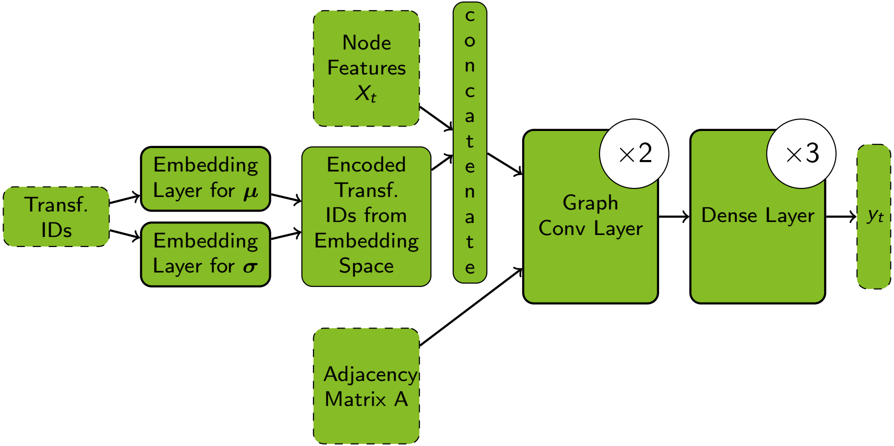
- 2 layers of graph convolutions
- each node aggregates the embeddings of its neighbour
- attention coefficient $\alpha$ weights each neighbour
- $\alpha_{u,v} = \text{softmax}\left(\frac{(W_3 h_u)^T (W_4 h_v W_5 e_{u,v})}{\sqrt{d_l}}\right)$

$$\underbrace{h_u^{(l+1)}}_{\text{node embedding}} = \sigma(\underbrace{W_1}_{\text{weights}} h_u^{(l)} + \sum_{v \in N(u)} \underbrace{\alpha_{u,v}}_{\text{attention coefficient}} W_2 h_v^{(l)})$$

## Our Approach



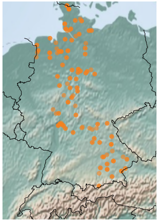

Node Features $X_t$
concatenate
Transf. IDs
Embedding Layer for $\mu$
Embedding Layer for $\sigma$
Encoded Transf. IDs from Embedding Space
Adjacency Matrix A
Graph Conv Layer
×2
Dense Layer
×3
$y_t$

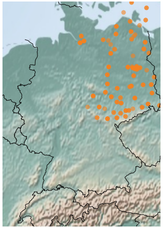

Node
Features
$X_t$
concatenate
Transf.
IDs
Embedding
Layer for $\mu$
Embedding
Layer for $\sigma$
Encoded
Transf.
IDs from
Embedding
Space
Graph
Conv Layer
×2
Dense Layer
×3
$y_t$
Adjacency
Matrix A

## Experiments: Dataset

Two datasets of German TSO

- approx. 175 transformers

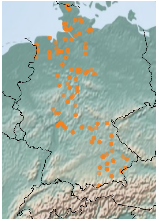


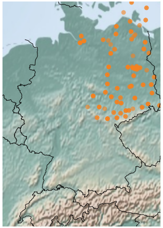


## Experiments: Dataset

Two datasets of German TSO

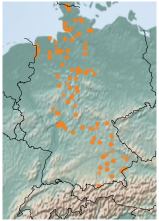
- approx. 175 transformers
- **Features**: weather, date/time, load and price forecast

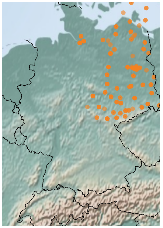




Two datasets of German TSO

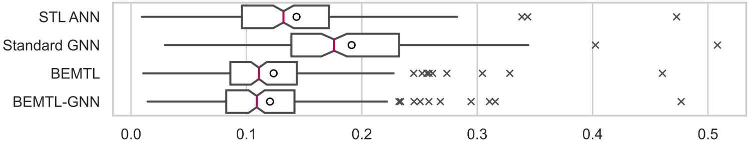
- approx. 175 transformers
- **Features**: weather, date/time, load and price forecast
- edges are defined by distance between transformers (distance 0km, 50km )





Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: Test RMSE




**STL**
Single Task Learning

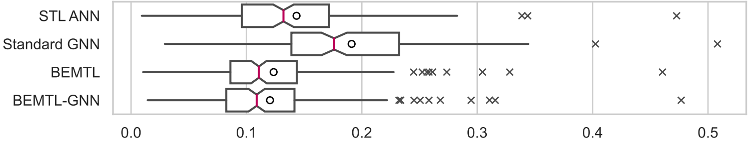
**Standard GNN**
no MT Embedding

**BEMTL**
MT Embedding, no GNN

**BEMTL-GNN**
GNN + MT Embedding

Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: Test RMSE




**STL**
Single Task Learning

**Standard GNN**
no MT Embedding

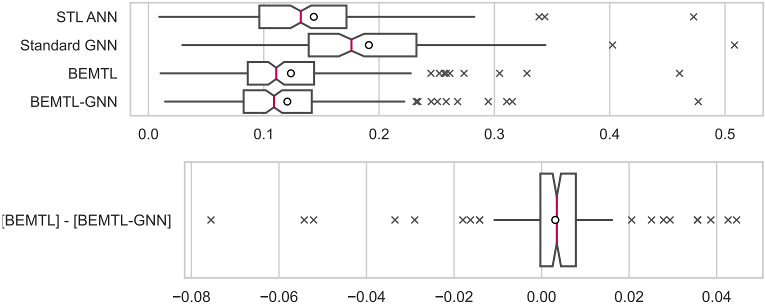
**BEMTL**
MT Embedding, no GNN

**BEMTL-GNN**
GNN + MT Embedding

**BEMTL GNN** achieves lowest average RMSE but only slight advantage over BEMTL

# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: Test RMSE




**STL**
Single Task Learning

**Standard GNN**
no MT Embedding

**BEMTL**
MT Embedding, no GNN

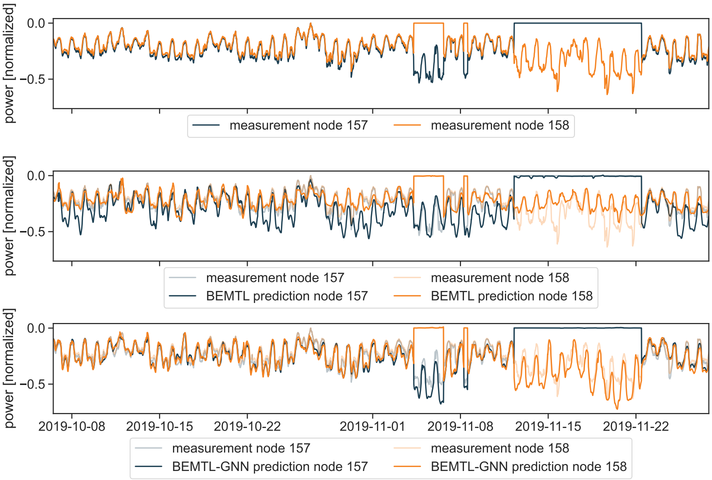
**BEMTL-GNN**
GNN + MT Embedding

**BEMTL GNN** achieves lowest average RMSE but only slight advantage over BEMTL

Comparing differences at each transformer, **BEMTL GNN** outperforms BEMTL on the majority of transformers (74%)

# Power flow forecasts at transmission grid nodes using GNNs

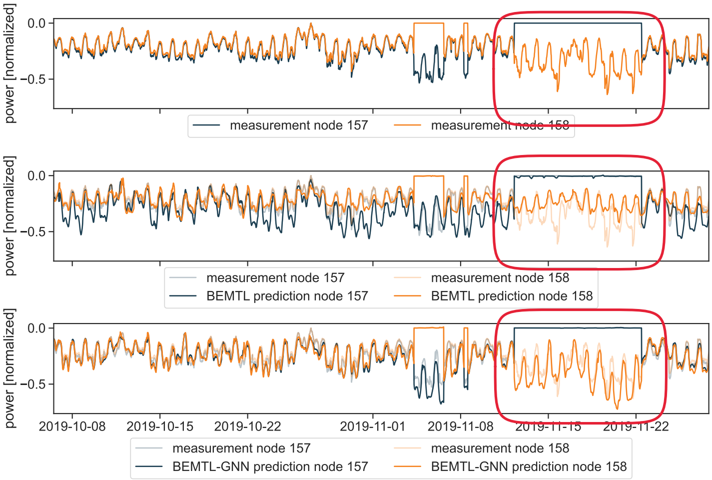
## Results Sparse Graph: BEMTL-GNN vs BEMTL



# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs BEMTL

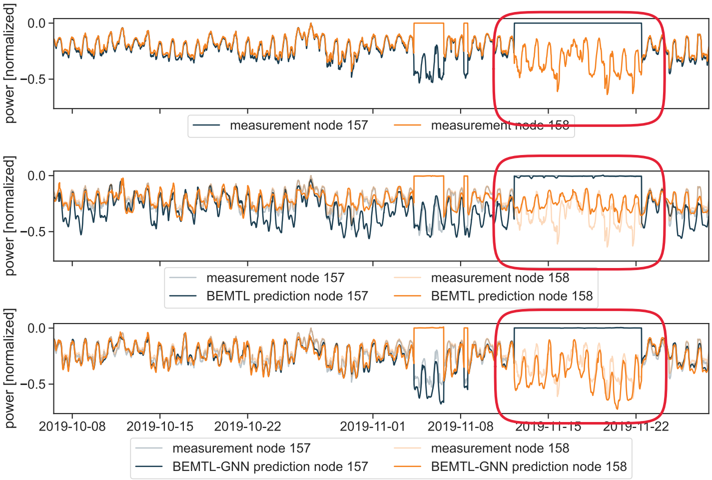
- Both models correctly predict inactivity



# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs BEMTL

- Both models correctly predict inactivity
- Only BEMTL GNN shows impact of the inactive transformer on the other

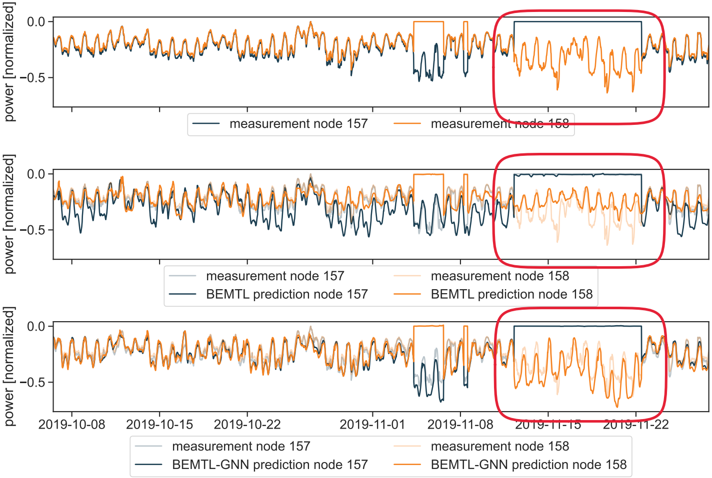


# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs BEMTL

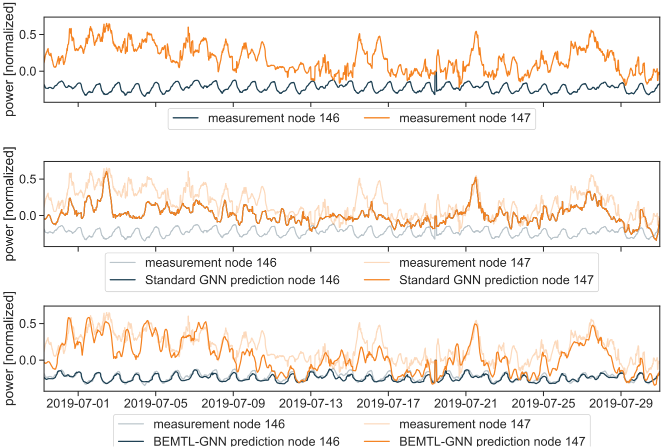
- Both models correctly predict inactivity
- Only BEMTL GNN shows impact of the inactive transformer on the other

→ BEMTL-GNN can indeed model relations between transformers



# Power flow forecasts at transmission grid nodes using GNNs

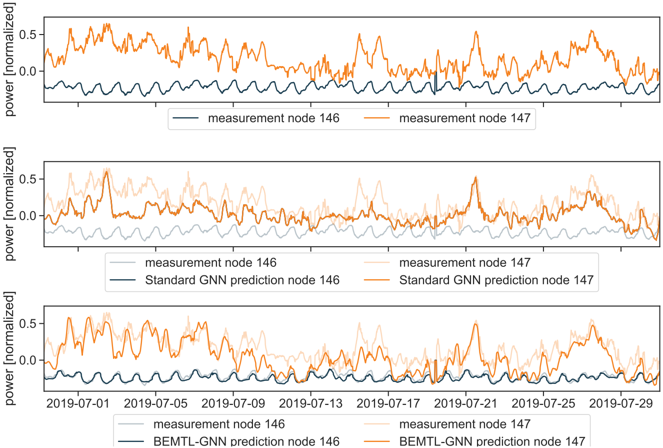
## Results Sparse Graph: BEMTL-GNN vs. GNN



# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs. GNN

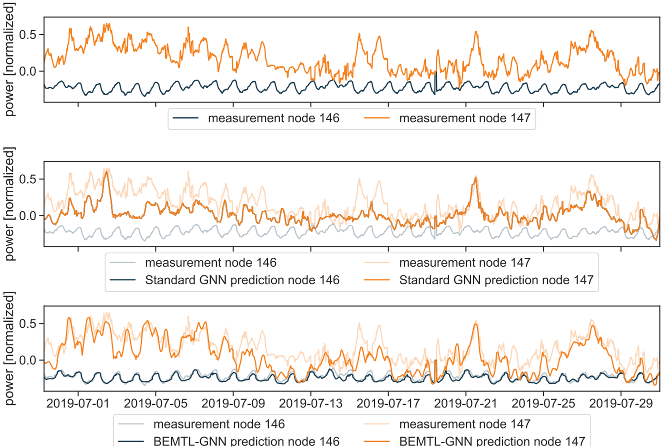
- Transformers share the same location



# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs. GNN

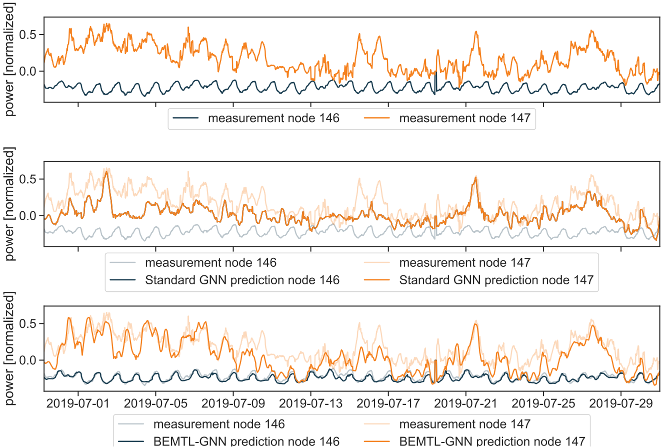
- Transformers share the same location
- GNN produces same output for both transformers



# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs. GNN

- Transformers share the same location
- GNN produces same output for both transformers
- BEMTL-GNN makes different predictions using the same input → much more accurate prediction for both nodes

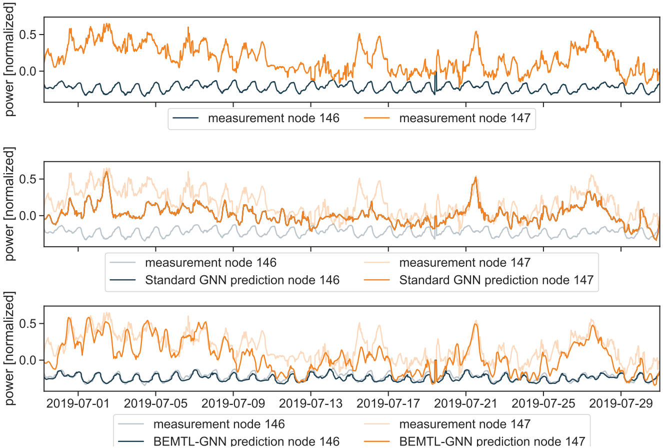


# Power flow forecasts at transmission grid nodes using GNNs

## Results Sparse Graph: BEMTL-GNN vs. GNN

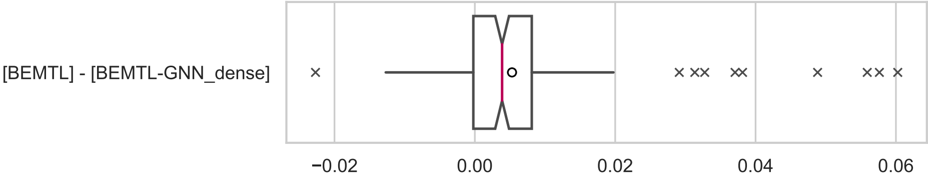
- Transformers share the same location
- GNN produces same output for both transformers
- BEMTL-GNN makes different predictions using the same input → much more accurate prediction for both nodes

→ The embedding is essential to model individual characteristics



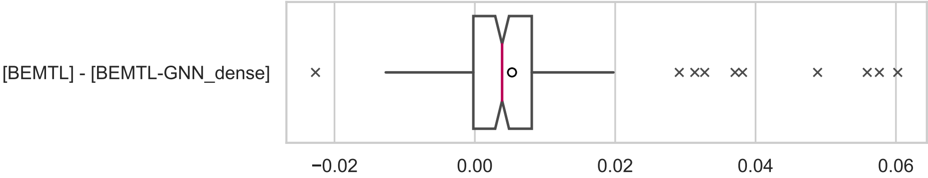
# Power flow forecasts at transmission grid nodes using GNNs

## Results Dense Graph



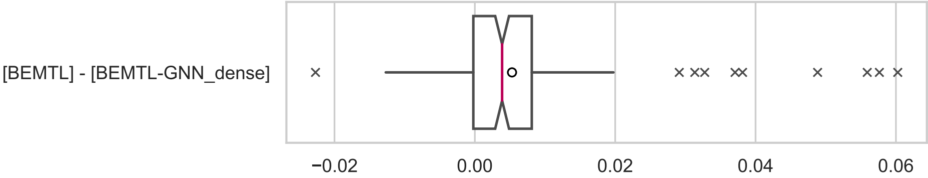
## Results Dense Graph

- RMSE is very similar to sparsely connected graph



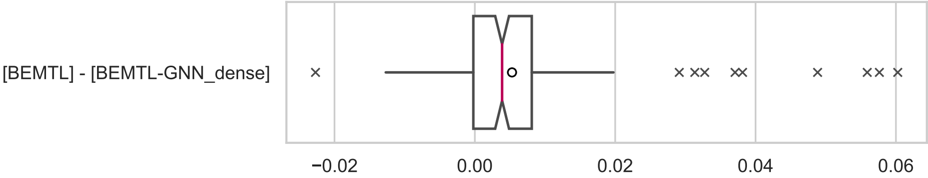
## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers



## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers
- interactions between transformers could not be observed as strongly?



## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers
- interactions between transformers could not be observed as strongly
  $\rightarrow$ two many factors in the neighbourhood aggregation?

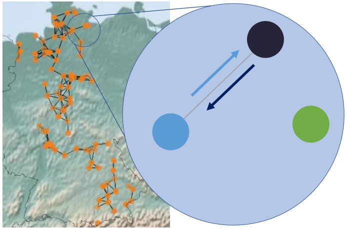
## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers
- interactions between transformers could not be observed as strongly
  $\rightarrow$ two many factors in the neighbourhood aggregation?
  $\rightarrow$ impact of further-away transformers not strong enaugh?

## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers
- interactions between transformers could not be observed as strongly
  → two many factors in the neighbourhood aggregation?
  → impact of further-away transformers not strong enaugh?

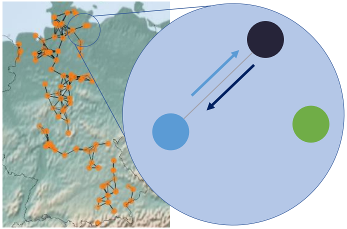
- Definition or Learning of Edges and edge weights, other GNN Layers, ...



# Power flow forecasts at transmission grid nodes using GNNs

## Results Dense Graph

- RMSE is very similar to sparsely connected graph
- BEMTL-GNN performs better than the BEMTL only on approx. 75% of the transformers
- interactions between transformers could not be observed as strongly
  $\rightarrow$ two many factors in the neighbourhood aggregation?
  $\rightarrow$ impact of further-away transformers not strong enaugh?



- Definition or Learning of Edges and edge weights, other GNN Layers, ...
- Data investigation: particularly interacting transformers

## Conclusion

- We combined a Multi-Task approach with an attention-based GNN to capture individual latent characteristics of transformers and their interactions

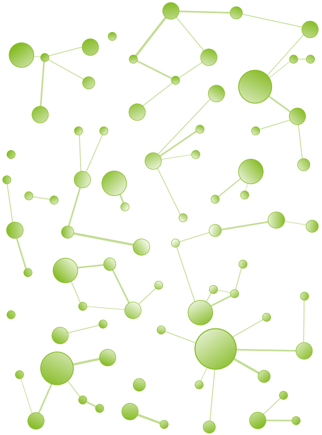
## Conclusion

- We combined a Multi-Task approach with an attention-based GNN to capture individual latent characteristics of transformers and their interactions
- For a sparsely connected graph this model shows best performance amongst compared models

- We combined a Multi-Task approach with an attention-based GNN to capture individual latent characteristics of transformers and their interactions
- For a sparsely connected graph this model shows best performance amongst compared models
- For a more densely connected, the influence of neighbouring transformers can be observed only to a small extent

- We combined a Multi-Task approach with an attention-based GNN to capture individual latent characteristics of transformers and their interactions
- For a sparsely connected graph this model shows best performance amongst compared models
- For a more densely connected, the influence of neighbouring transformers can be observed only to a small extent

$\rightarrow$ Experiments on sparse graph as Proof of concept



# FDGNN: Fully Dynamic Graph Neural Network

Alice Moallemy-Oureh, Silvia Beddar-Wiesing

# FDGNN: Fully Dynamic Graph Neural Network[7]

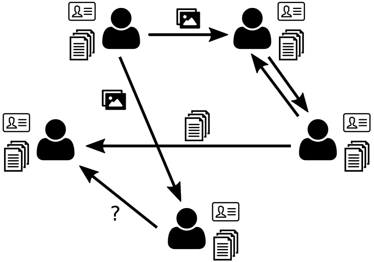
## Motivation

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

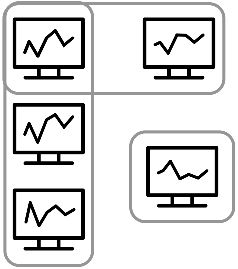
## Motivation

**Social Networks**




- Recommender system (link prediction)
- Fraud detection (node classification)
- ...

**Stock Market Prices**



- Stock price prediction (attribute prediction)
- ...

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Motivation

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

**Current Work:**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

**Current Work:**

- Most of the GNNs in literature can only handle (unattributed) **growing graphs**.
  - ⟹ Deletions of nodes are not modeled
  - ⟹ Nodes and edges do not necessarily have attributes
  - ⟹ Attributes are not dynamic as well

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Motivation

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

**Current Work:**

- Most of the GNNs in literature can only handle (unattributed) **growing graphs**.
  - ⟹ Deletions of nodes are not modeled
  - ⟹ Nodes and edges do not necessarily have attributes
  - ⟹ Attributes are not dynamic as well
- Neural Spatio-Temporal Point Processes is an upcoming field, but only few approaches address dynamic graphs and mainly focus on the graph's edges.

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Motivation

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

**Current Work:**

- Most of the GNNs in literature can only handle (unattributed) **growing graphs**.
  - ⟹ Deletions of nodes are not modeled
  - ⟹ Nodes and edges do not necessarily have attributes
  - ⟹ Attributes are not dynamic as well
- Neural Spatio-Temporal Point Processes is an upcoming field, but only few approaches address dynamic graphs and mainly focus on the graph's edges.
- Most of the models address only **Link Prediction** and **Event Time Prediction**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Motivation

⇒ Graphs dynamic in **structure** and **attributes**

⇒ Tasks: Node classification, link prediction, graph classification, event prediction, ...

**Current Work:**

- Most of the GNNs in literature can only handle (unattributed) **growing graphs**.
  - ⟹ Deletions of nodes are not modeled
  - ⟹ Nodes and edges do not necessarily have attributes
  - ⟹ Attributes are not dynamic as well
- Neural Spatio-Temporal Point Processes is an upcoming field, but only few approaches address dynamic graphs and mainly focus on the graph's edges.
- Most of the models address only **Link Prediction** and **Event Time Prediction**
- **Attribute Prediction** mostly of nodes can be just found for GNNs working on descrete-time and without any structural changes of the graph.

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Motivation

The FDGNN is capable of processing

- both **structural** and **attribute dynamics** and
- the efficient graph representation as **graph streams**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

The FDGNN is capable of processing

- both **structural** and **attribute dynamics** and
- the efficient graph representation as **graph streams**

and learns

- an **expressive vector representation** of the graph incl. its attributes and
- functions representing the **temporal evolution** of the graph

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

The FDGNN is capable of processing

- both **structural** and **attribute dynamics** and
- the efficient graph representation as **graph streams**

and learns

- an **expressive vector representation** of the graph incl. its attributes and
- functions representing the **temporal evolution** of the graph

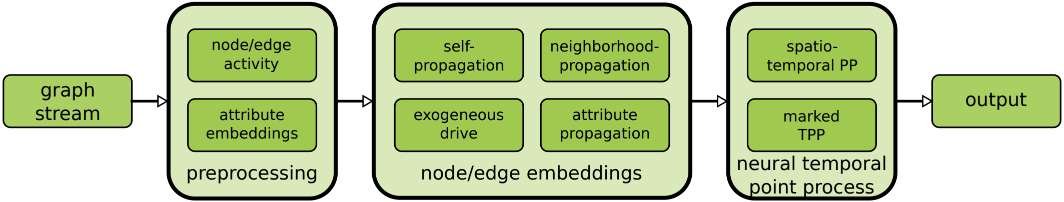
to address potentially different learning problems, such as

- event and event time prediction or
- node/edge/graph classification/regression.

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## FDGNN Architecture



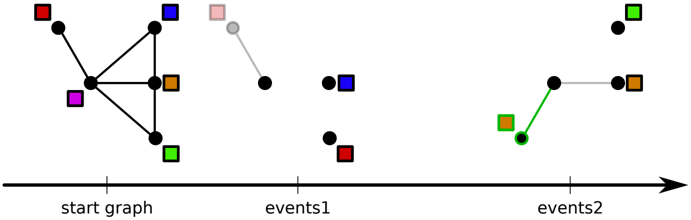

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Input: Graph Stream

graph stream

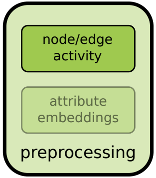
- Start graph and stream of different **events**
- **Structural** changes: addition/deletion of nodes or edges
- **Attribute** changes of nodes or edges
- not necessarily **equidistant time**




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Preprocessing: Node/Edge Activity




- Activity encodes **existence** of node and edges at a time

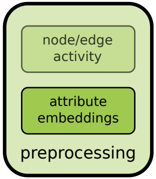
$$\text{active node} \longmapsto 1$$
$$\text{inactive node} \longmapsto 0$$

- Thereby, the **deletion behavior** can also be learned afterwards

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Preprocessing: Attribute Embedding

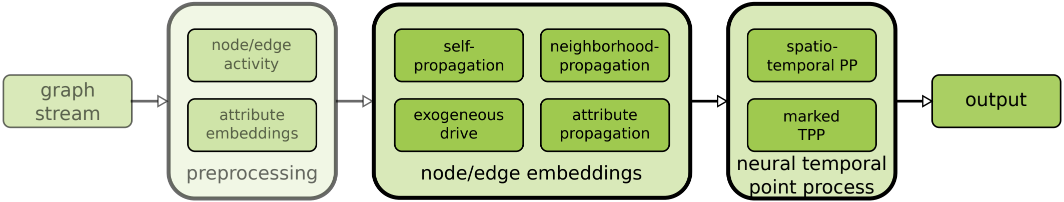



- Vector representation of the node or edge textbfattributes
- Attribute embedding as **preprocessing**
- **Depending on dataset** considering a suitable attribute embedding into the $\mathbb{R}^n$ (e.g., word2vec)

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

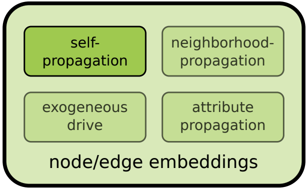
## FDGNN Architecture




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

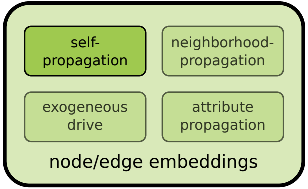
## Embedding: Self-Propagation



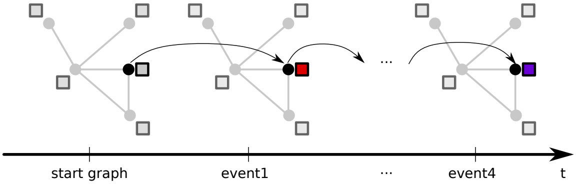

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding: Self-Propagation




- includes temporal evolution of the current node/edge embedding
- with integrated forgetting via **temporal attention** (GATv2)




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding: Neighborhood-Propagation

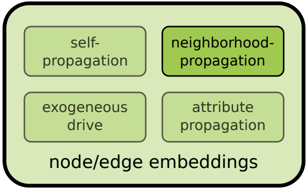
| | |
|---|---|
| self-propagation | **neighborhood-propagation** |
| exogeneous drive | attribute propagation |

node/edge embeddings

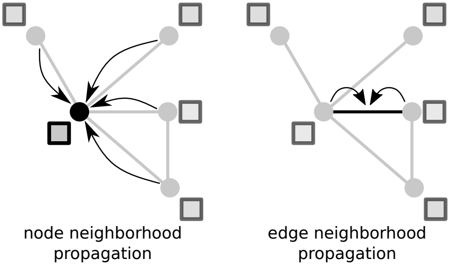
[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding: Neighborhood-Propagation



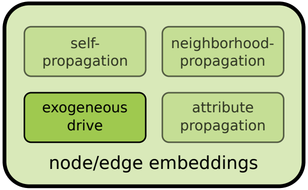

- Cumulates **local neighborhood information** in the graph (seperately for nodes and edges)
- Classical Graph Attention Neural Network (GATv2, without self-loops)




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

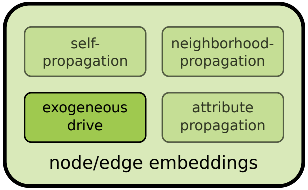
## Embedding: Exogeneous Drive



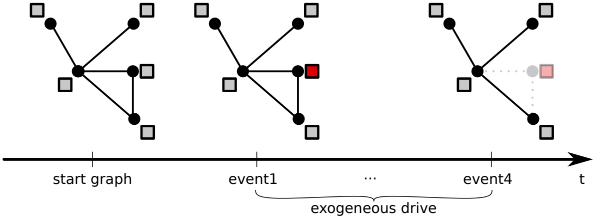

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding: Exogeneous Drive



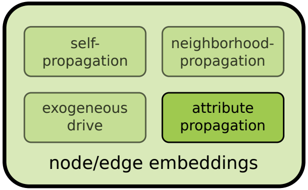

- **time interval** between current event and the last event on the same node/edge




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

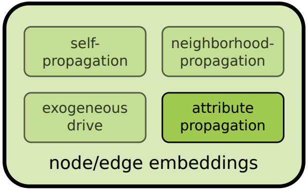
## Embedding: Attribute Propagation




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding: Attribute Propagation

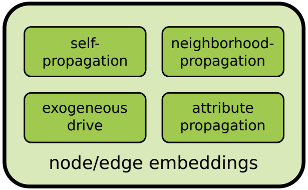



- Encodes temporal evolution of node/edge attributes
- Recurrent Layer

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

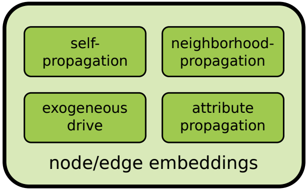
## Embedding




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Embedding

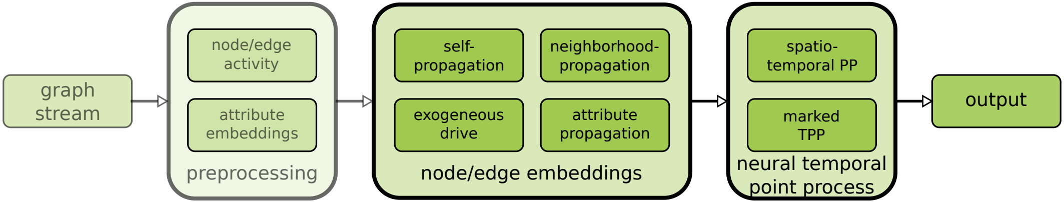



- the event embedding is then determined by the sum of the modules
- passed through an activation function
- one embedding vector for each node and edge seperately

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

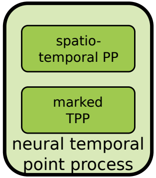
## FDGNN Architecture



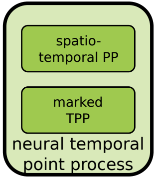

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Temporal Point Process (TPP)



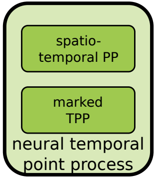

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469




### Temporal Point Process (TPP):

- **probabilistic generative model** for continuous-time event sequences

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469



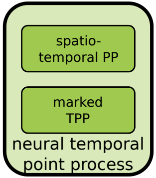

### Temporal Point Process (TPP):

- **probabilistic generative model** for continuous-time event sequences
- can model specific **temporal pattern** in variable-length event sequences

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

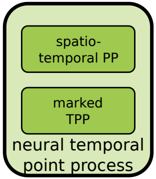
## Temporal Point Process (TPP)




### Temporal Point Process (TPP):

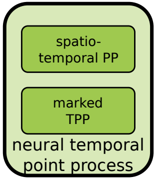
- **probabilistic generative model** for continuous-time event sequences
- can model specific **temporal pattern** in variable-length event sequences
- conditional probability over time is often defined via **conditional intensity functions** considering the history
- intensity functions represent number of events over time

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469




**Neural TPP:**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469



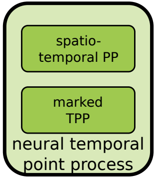

**Neural TPP:**

- extends TPPs to the Deep Learning approach
- learns intensity functions with Neural Networks
- allows for **learning more complex temporal pattern**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

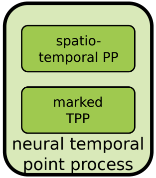
## Marked Spatio-Temporal Point Process




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Marked Spatio-Temporal Point Process

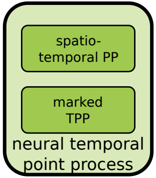



- **space** is determined by location in graph (node/edge)
- **marks** (additional event information) correspond to node/edge attributes
- intensity function is the product of spatio-temporal and mark intensities

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

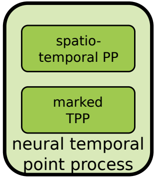
# FDGNN: Fully Dynamic Graph Neural Network[7]

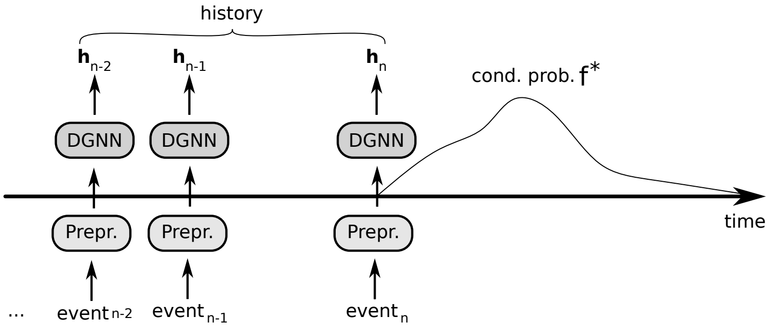
## FDGNN: Marked Neural Spatio-Temporal Point Process




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

Marked Neural Spatio-Temporal Point Process models pattern in attributed dynamic graph stream




[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

# FDGNN: Fully Dynamic Graph Neural Network[7]

## Training and Prediction

- Update the parameter set by, e.g., **maximizing the likelihood** of observed events and
- minimizing the intensity of unobserved events (**survival probability**)
- loss function is approximated by **Monte Carlo Sampling**
- **predictions** can be directly inferred using the probability function

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

- FDGNN processes **dynamic graphs** with structural and attribute changes
- preprocessing enables handling of **attributes** of arbitrary type and learning of **deletions**
- the embedding module considers the entire **complex information**
- finally, the history of embeddings in the TPPs is processed to make various **predictions**

[7] Moallemy-Oureh, Beddar-Wiesing, Nather, Thomas: *FDGNN: Fully Dynamic Graph Neural Network*, arXiv:2206.03469

GAIN

# Thank you for your attention!

# Questions?

P.S.: We are looking for new colleagues :)

Intelligent Embedded Systems

UNIKASSEL VERSITÄT

Fraunhofer IEE

Bundesministerium für Bildung und Forschung