Graph Neural Networks for Electrical Grid State Estimation

Armin Saur

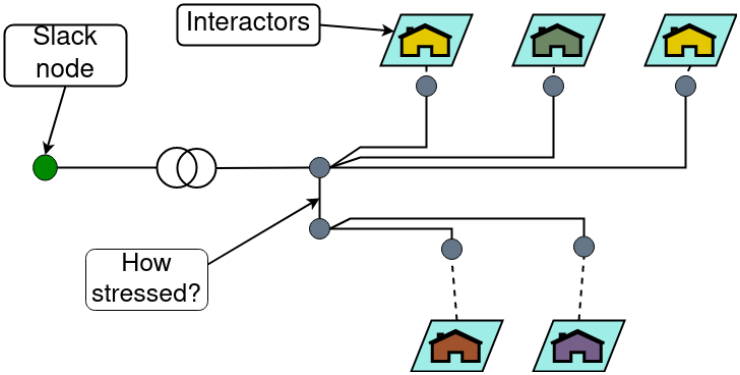
Freiburg i. Br., 25. March 2024

First Examiner:	Prof. H. Bast
Second Examiner:	Prof. C. Wittwer
Advisor:	B. Rückauer, S. Walter and W. Biener,

Problem

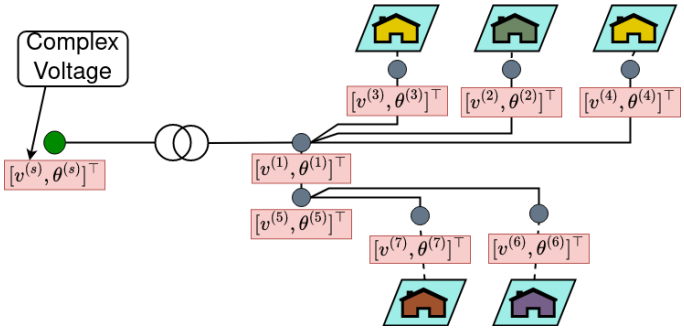
Given Low-voltage Network

An example *Small* LVN



Given and Target

- Complex Power $\underline{S}$
  - $\underline{S}_i = P_i + j\,Q_i$
  - $\forall i, i \in \mathcal{N}$
- Edge admittance
  - $\underline{Y}_{i,k} = G_{i,k} + j\,B_{i,k}$
  - $\forall (i,k), (i,k) \in \mathcal{E}$
- Complex Voltage $\underline{V}$
  - $\underline{V}_i = |V_i| \cdot \exp(j\,\theta_i)$
  - $\forall i, i \in \mathcal{N}$



State Estimation Task

Derive with *Kirchhoff's circuit law* for $i \in \mathcal{N}$

$$P_i + j\,Q_i = |V_i|\, e^{j\,\theta_i} \cdot \Big(\sum_{k \in A(i)} \big(G_{i,k} + j\,B_{i,k} \big) \cdot |V_k|\, e^{\theta_k} \Big)^*$$

$$\forall i \in \mathcal{N}$$

- (Apply some *Non-linear solver*)
- Related to network stability and reliability

What if not all P and Q are known?

Bonus

- Edge admittance [Ω^{-1}]
  - Shunt conductivity and Shunt susceptance
  - $\underline{Y} = G + j\,B$
- Edge Impedance [Ω]
  - Series reactance and Series resistance
  - $\underline{Z} = R + j\,X$

- Substitute complex power and complex current
  $\underline{Y} = \frac{1}{\underline{Z}} = \frac{\underline{I}}{\underline{V}}$
- *Kirchhoff's circuit law*
  $\underline{I}_i = \sum_{k \in \boldsymbol{A}(i)} \underline{I}_k$
- Grid state matrix $\boldsymbol{Y}_t$
  $\boldsymbol{y}_i = [\,|V_i|\,,\,\theta_i\,]^\top$ and
  $\boldsymbol{Y}_t = [\,\boldsymbol{y}_{t,1}\,,\ldots,\,\boldsymbol{y}_{t,\mathcal{N}}\,]^\top$

Bonus2

- State Estimation Task

$$0 = -P_i + \sum_{k=1}^{\mathcal{N}} |V_i| \ |V_k| \cdot \left(G_{i,k} \cos\left(\triangle \theta_{i,k}\right) + B_{i,k} \sin\left(\triangle \theta_{i,k}\right)\right)$$

$$0 = -Q_i + \sum_{k=1}^{\mathcal{N}} |V_i| \ |V_k| \cdot \left(G_{i,k} \cos\left(\triangle \theta_{i,k}\right) + B_{i,k} \sin\left(\triangle \theta_{i,k}\right)\right)$$

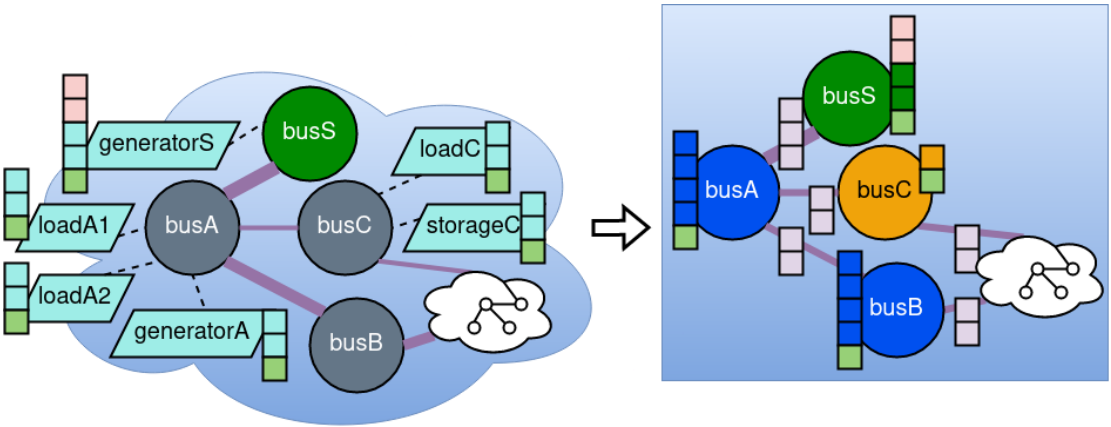
- $\forall i, i \in \boldsymbol{\mathcal{N}}$
- Complexity $\mathcal{O}(2(\mathcal{N}-1))$

Solution

Heterogeneous Graph

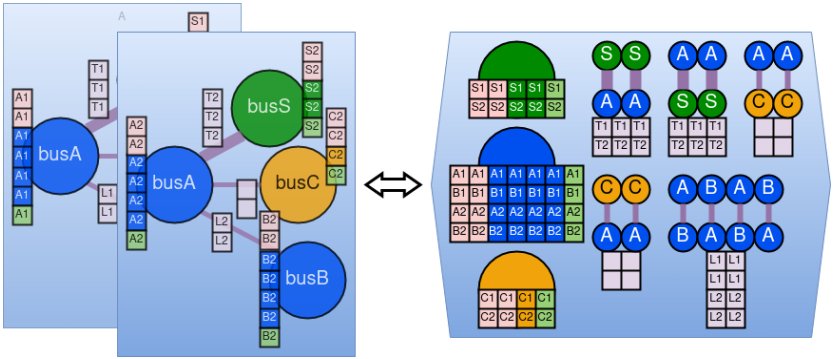
(left) LVN

(right) Heterogeneous Graph



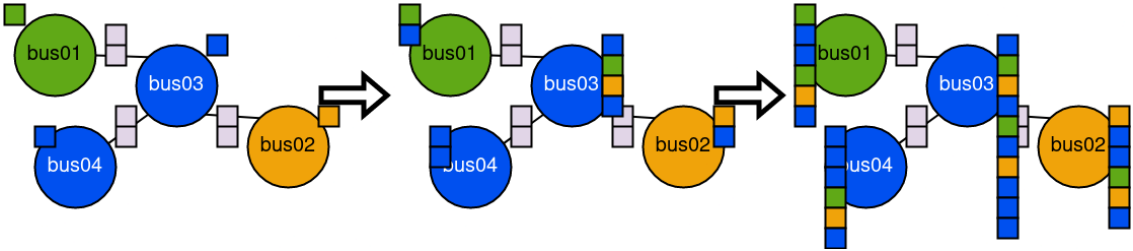
Data structure

- Node-Edge-Node Types: $\boldsymbol{R} \subset \boldsymbol{R}^{(\mathcal{N})} \times \boldsymbol{R}^{(\mathcal{E})} \times \boldsymbol{R}^{(\mathcal{N})}$



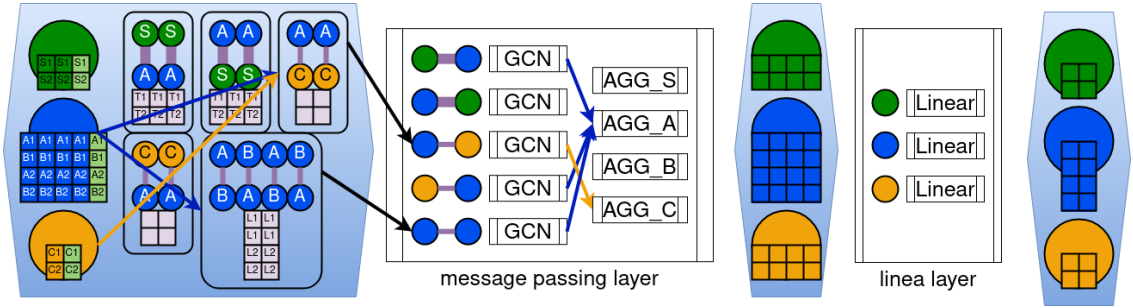
Graph Convolution Network

- Message-passing framework $f_{\text{msg}}^{(\cdot)}$
- Concatenation aggregation $\bigoplus_{j \in \boldsymbol{A}(i)}^{(\text{cc})}(\cdot)$ and update $\lambda(\cdot)$
- Target node $i \in \mathcal{N}$



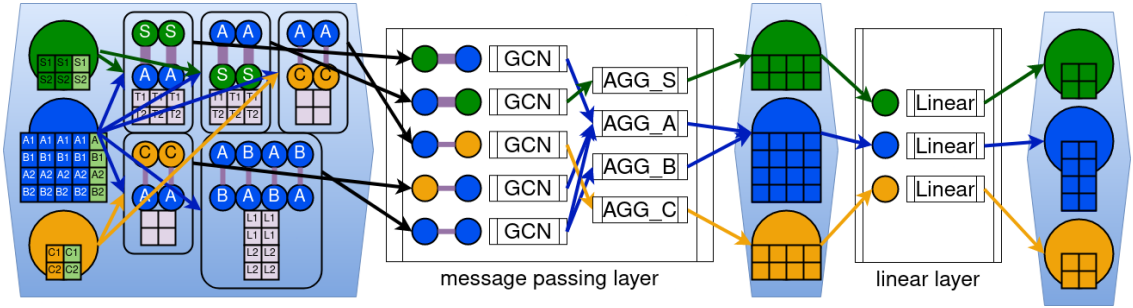
Heterogeneous Graph Convolution Network

- Message-passing framework map $\{f_{\text{msg}}^{(\text{l,r})} : \forall \boldsymbol{r} \in \boldsymbol{R}\}$
- General layer map $\{f^{(\text{l,r}^{(\mathcal{N})})} : \forall \boldsymbol{r}^{(\mathcal{N})} \in \boldsymbol{R}^{(\mathcal{N})}\}$



Heterogeneous Graph Convolution Network

- Message-passing framework map $\{f_{\text{msg}}^{(\text{l,r})} : \forall \boldsymbol{r} \in \boldsymbol{R}\}$
- General layer map $\{f^{(\text{l,r}^{(\mathcal{N})})} : \forall \boldsymbol{r}^{(\mathcal{N})} \in \boldsymbol{R}^{(\mathcal{N})}\}$



Approach GSETR

- Use 3 *Gatv2 Convolution* layer
  - Can learn edge features G, B
- No normalization
- Analyze former work
  - Avoid bias parameter weights
  - Avoid batch normalization
  - Clarify structure and fix bugs

- Data transformation
  - Weighted loss for $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$

$$|V_i| = 400 \cdot |V_i|$$

$$|V_i| = \frac{180}{\pi} \cdot \theta_i$$

- Relative Root Mean Squared Error
- Supervised: Apply non-linear solver

Bonus

- Concatenation aggregation

$$\begin{aligned} \boldsymbol{m}_i^{(l)} &= \bigoplus_{j \in \boldsymbol{A}(i)}^{(cc)} (\boldsymbol{h}_k^{(l)}) \\ &= \|_{k \in \boldsymbol{A}(i)} \ \boldsymbol{h}_j^{(l)} \end{aligned}$$

- Concatenation update function

$$\begin{aligned} \boldsymbol{h}_i^{(l)} &= \lambda(\boldsymbol{h}_i^{(l-1)}, \boldsymbol{m}_i^{(l)}) \\ &= [\boldsymbol{h}_i^{(l-1)} \parallel \boldsymbol{m}_i^{(l)}]^\top. \end{aligned}$$

Bonus2

- Edge score

$$s_{i,k} = \boldsymbol{a}^{\top} \sigma_{\text{att}}(\boldsymbol{W}_{\text{att}}^{\top(\text{l})}[\boldsymbol{h}_i^{(\text{l})} \parallel \boldsymbol{h}_k^{(\text{l})}] + \boldsymbol{W}_{\mathcal{E}\text{att}}^{(\text{l})} \boldsymbol{x}_{t,(i,k)}^{\mathcal{E}})$$

- Normalized edge score

$$\alpha_{i,j} = \frac{\exp(s_{i,j})}{\sum_{k \in \boldsymbol{A}(i)} \exp(s_{i,k})}$$

- Update hidden values

$$\boldsymbol{h}_i^{(\text{l}+1)} = \alpha_{i,i} \boldsymbol{W}_{\text{att}}^{(\text{l})} \boldsymbol{h}_i^{(\text{l})} \sum_{j \in \boldsymbol{A}(i)} \alpha_{i,j} \boldsymbol{W}_{\text{att}}^{(\text{l})} \boldsymbol{h}_j^{(\text{l})}$$

Bonus 3

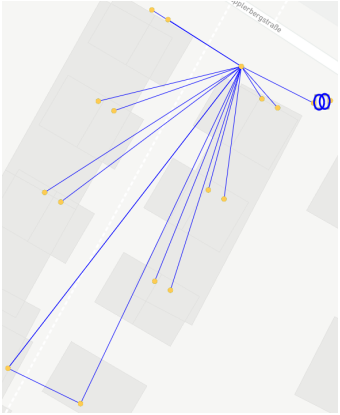
RRMSE

$$L = \sqrt{\frac{\frac{1}{\mathcal{T}\mathcal{N}T}\sum_{t=1}^{T}\sum_{i=1}^{\mathcal{N}}\sum_{\psi=1}^{\mathcal{T}}(y_{t,i}^{(\psi)} - \hat{y}_{t,i}^{(\psi)})^2}{\sum_{t=1}^{T}\sum_{i=1}^{\mathcal{N}}\sum_{\psi=1}^{\mathcal{T}}(y_{t,i}^{(\psi)})^2}}$$

Results

Setup

- Assign types
  - *slack*, *measured* and *unmeasured*
- Random $k = 9$ *unmeasured* nodes
  - Drop half of information
- Repeat three times
  - Three different type profiles
- Create $m = 9$ models of GSETR
- Three models for each type profile

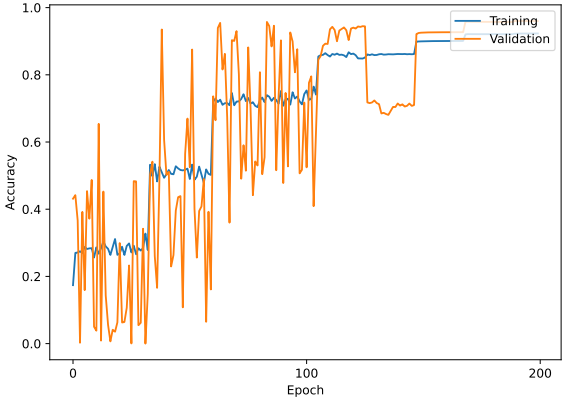


LVN *Allensbach*

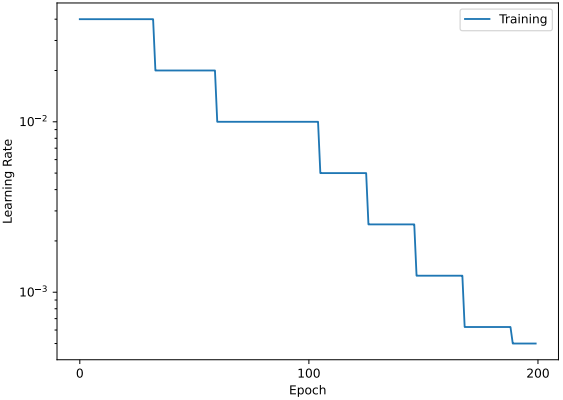
Accuracy

Phase	mean accuracy	standard deviation accuracy
Validation	91.0 %	3.2 %
Evaluation	91.6 %	3.6 %

(1) Trainig and Validation

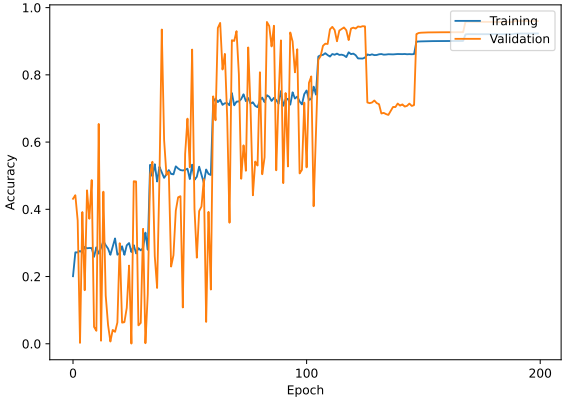


Accuracy per epoch

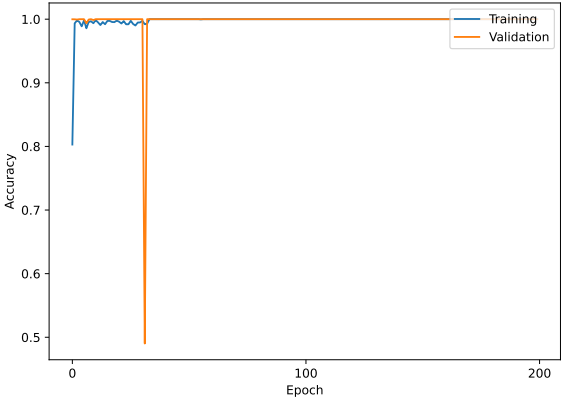


Learning rate per epoch

(1) Trainig and Validation per Attribute

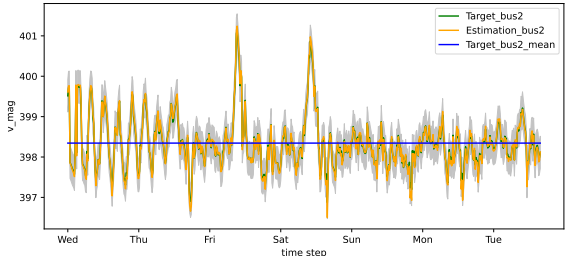


Voltage magnitude accuracy per epoch

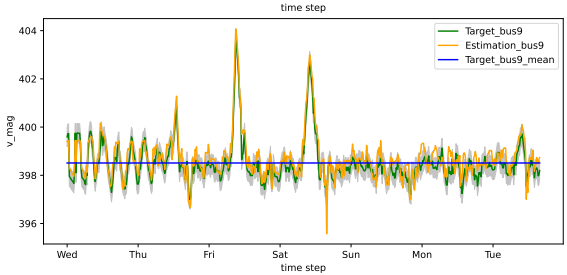


Voltage angle accuracy per epoch

(1) Evaluation time series voltage magnitude

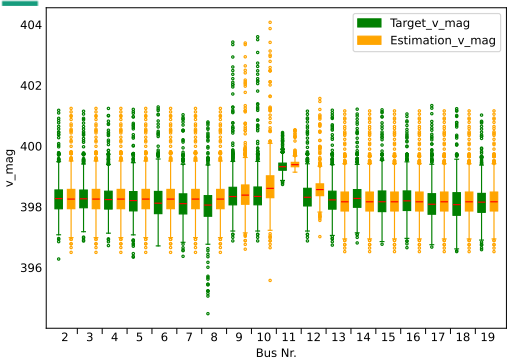


Voltage magnitude best node

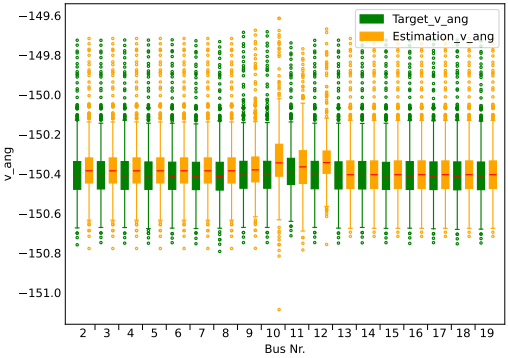


Voltage magnitude worst node

(1) Evaluation per node



Voltage magnitude



Voltage angle

Limitations validation

GSETR applied on only one graph topology

- Stable on heterogeneous graph topology!
- Flat topology
- Evaluate stability on different LVN topologies.
- Define realistic type assignment.
- Define additional metrics.