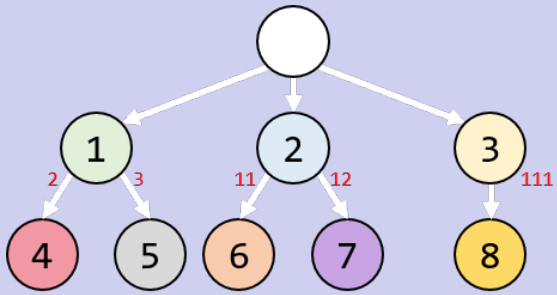


Master Thesis Seminar Talk

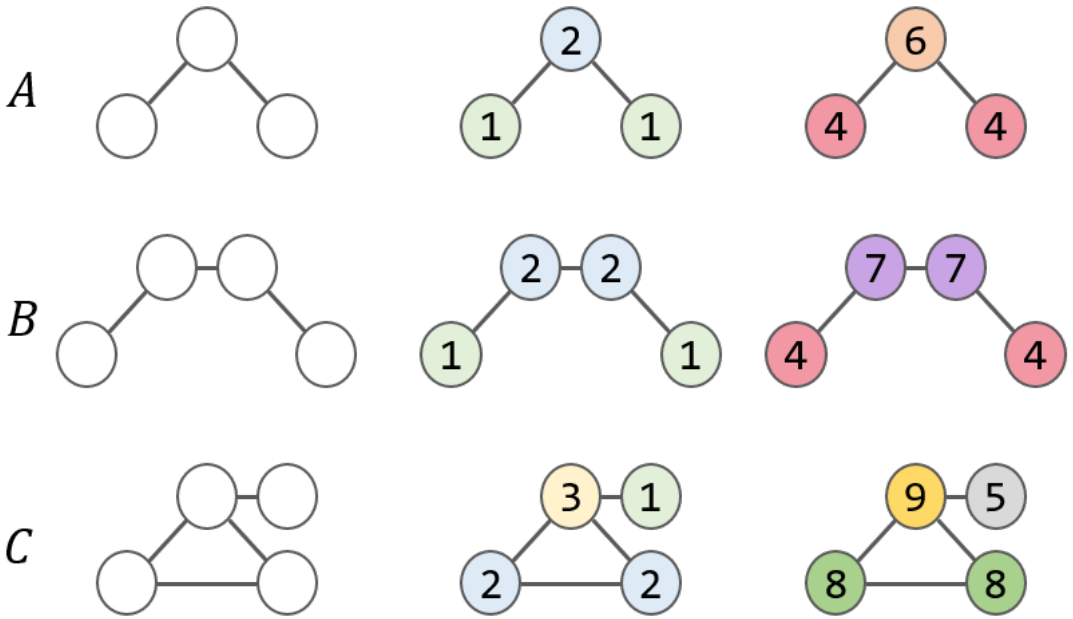
Progress Update

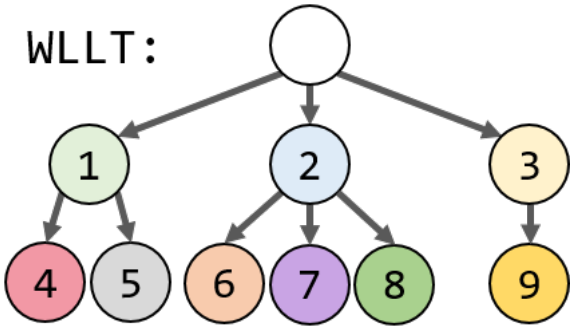
Fabrice Beaumont

Department of Information Systems and Artificial Intelligence - **Dr. Pascal Welke**

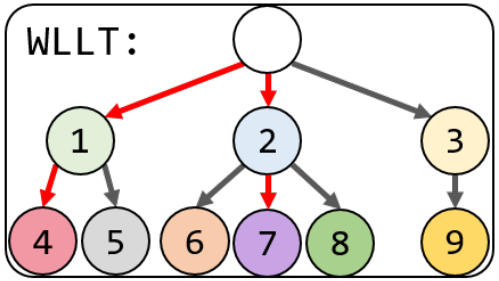
14. September 2022

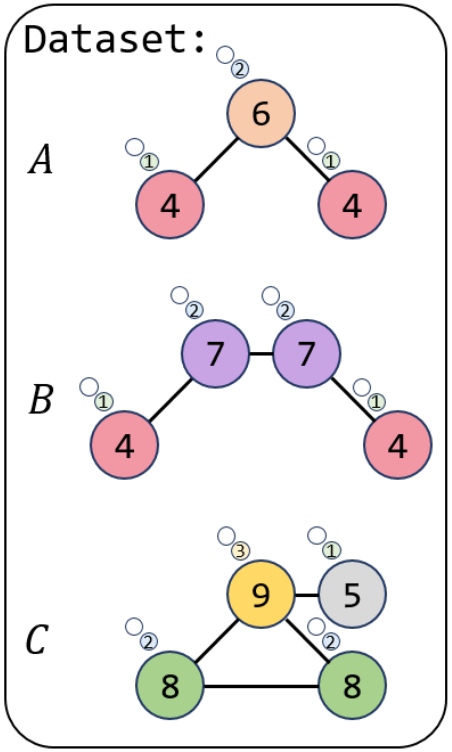
Example of the whole procedure



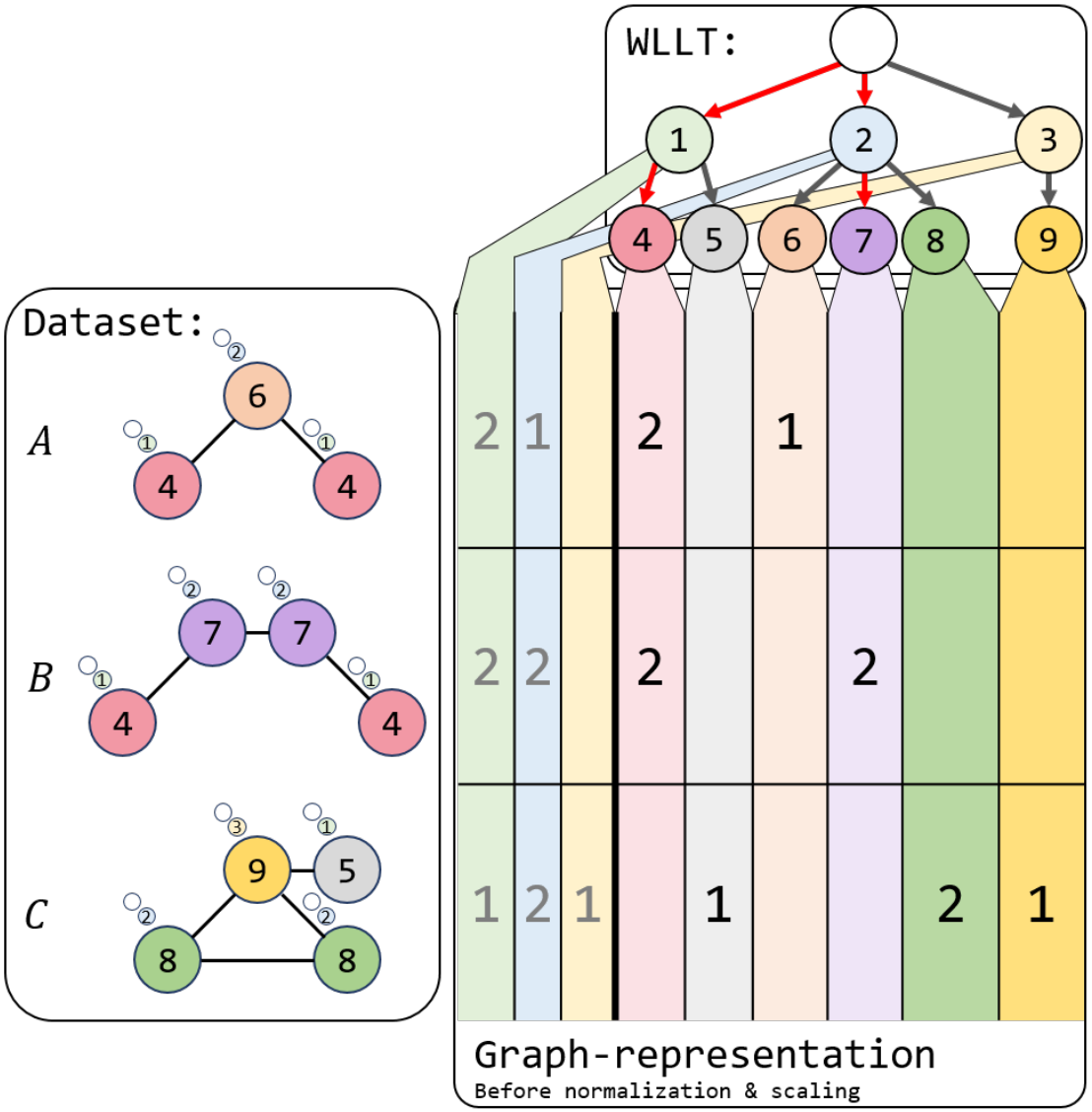


Example of the whole procedure

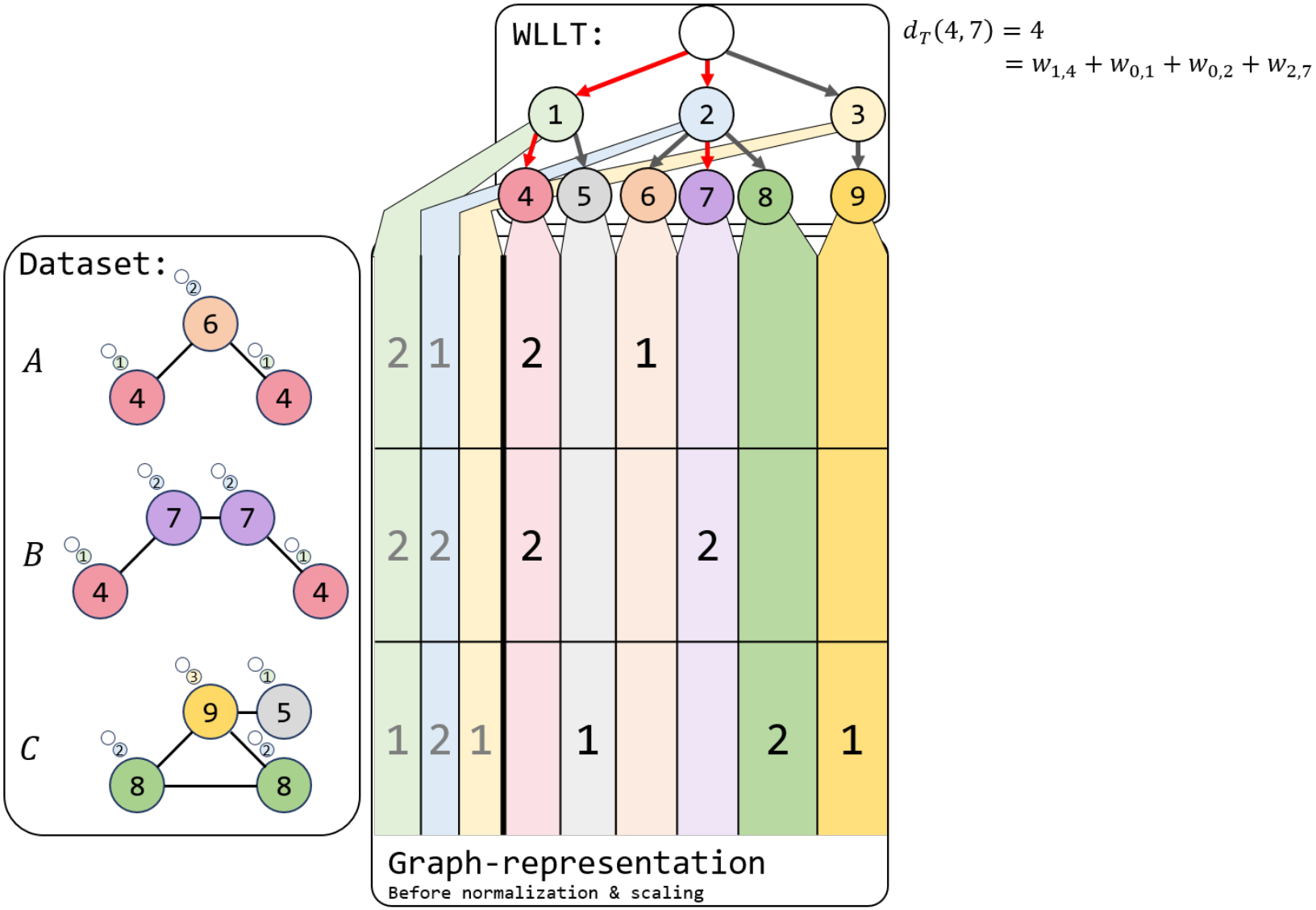




Example of the whole procedure

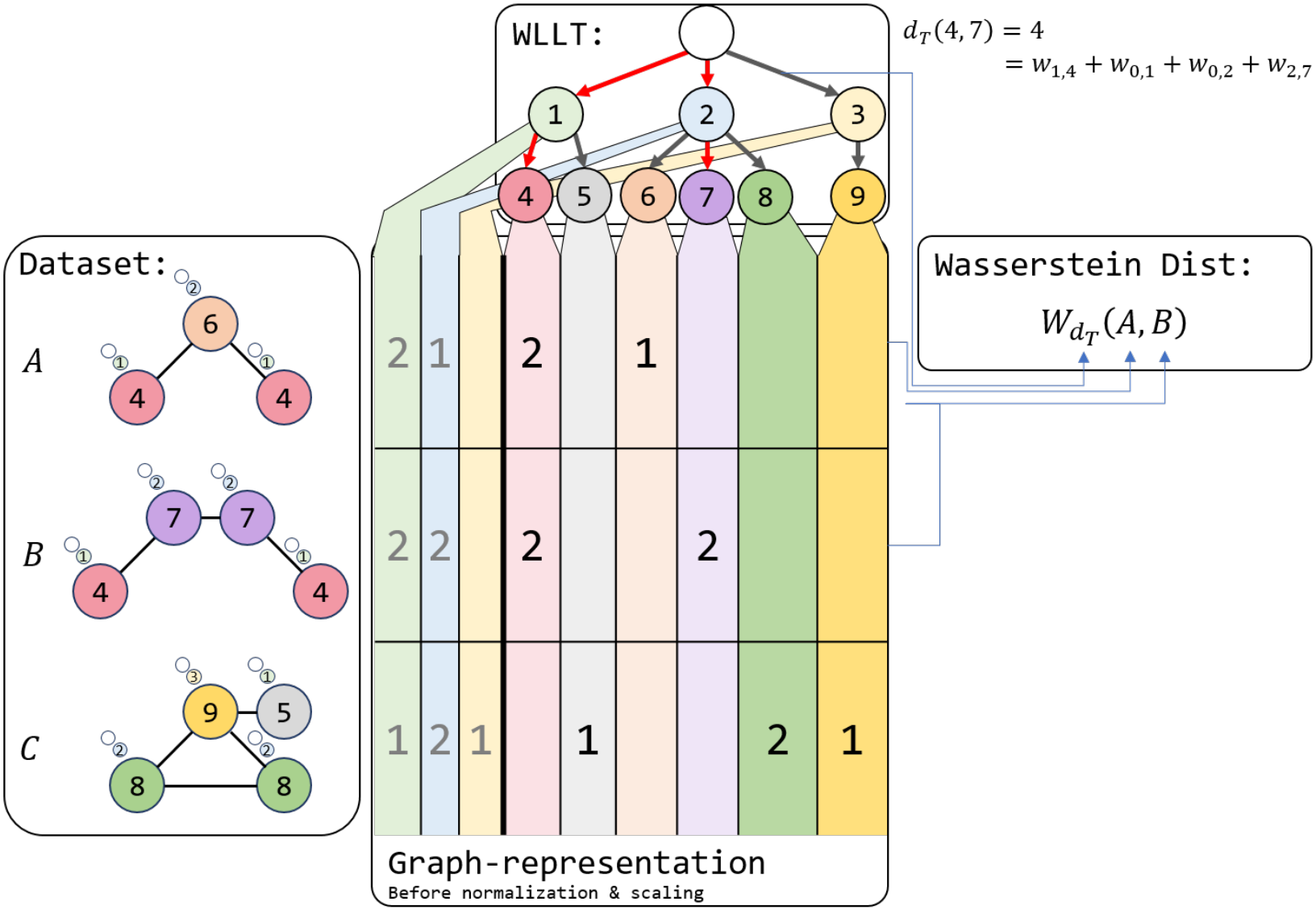


Example of the whole procedure

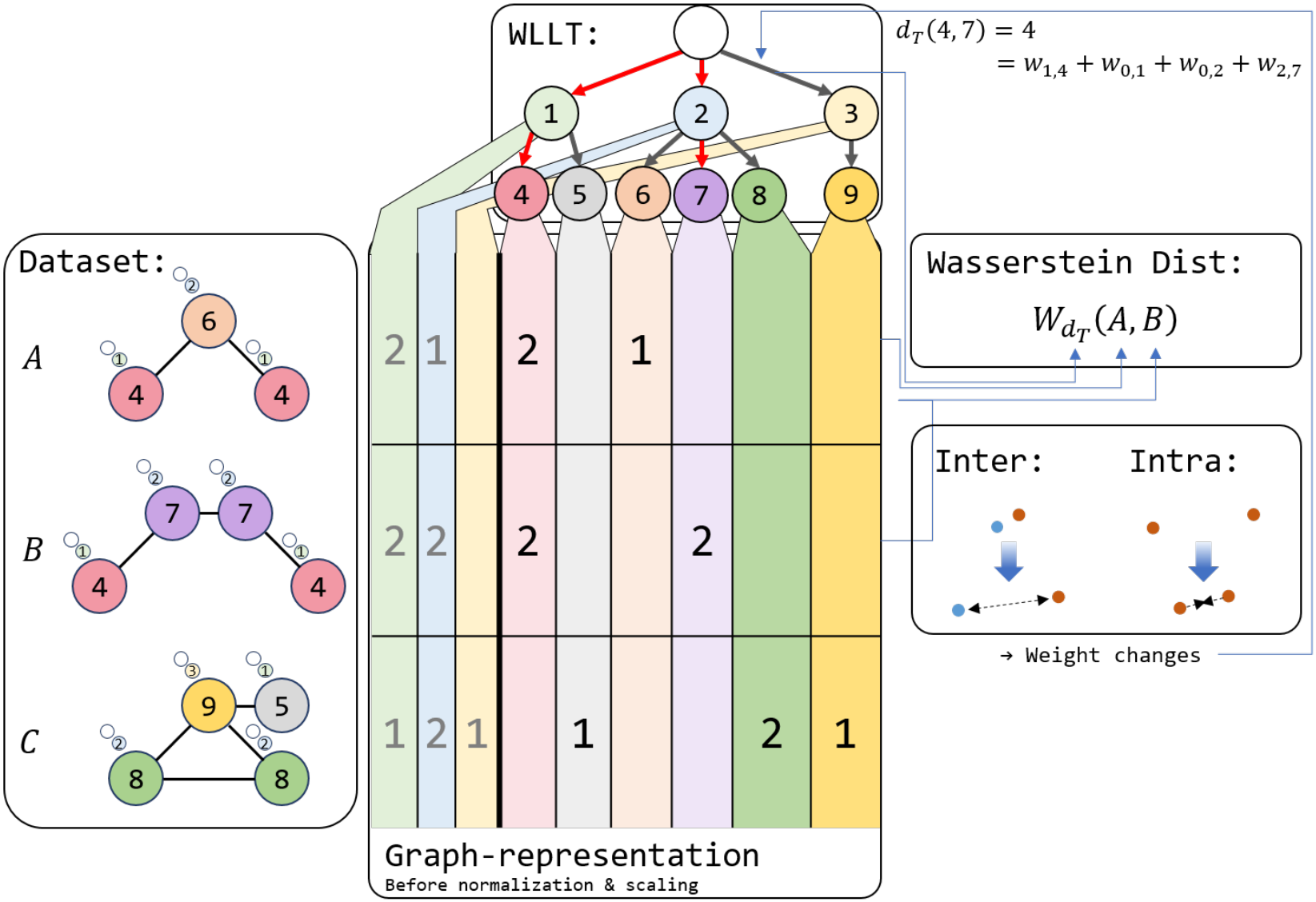


$$d_T(4,7) = 4$$
$$= w_{1,4} + w_{0,1} + w_{0,2} + w_{2,7}$$

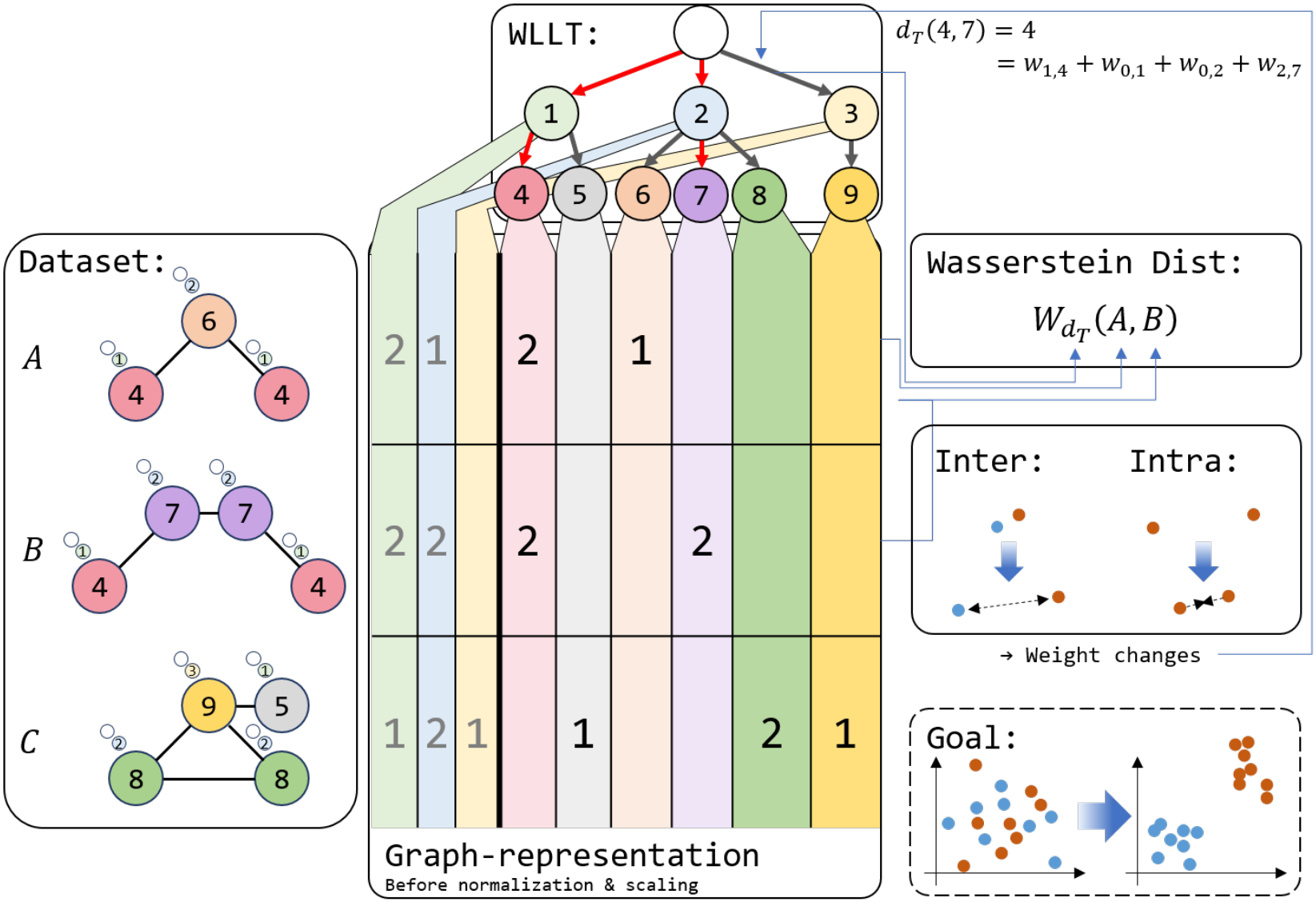
Example of the whole procedure



Example of the whole procedure



Example of the whole procedure



Meta Parameters

Procedure

- WLLT depth
- Update scope (w.r.t. layers)
- Update intensity: Push and Pull factors (fixed or percentage)
- Adjust weights to leaves only
- Heaviest earth threshold

Machine Learning

- Update frequency (batch size)
- Learning rate
- Adjust weights to leaves only
- Class imbalance factor

First results

- Clustering scores improve reliably, after enough iterations or high enough WLLT depth.
- WLLT layers 1, 2 and 3 get the most total weight increases.
- WLLT layer 2 grows more rapidly than all others.
-
- TODO: Track number not intensity of weight updates per Layer?

! Switch to go from relative to absolute push/pull factors

- OGB datasets

Current todos / Outlook

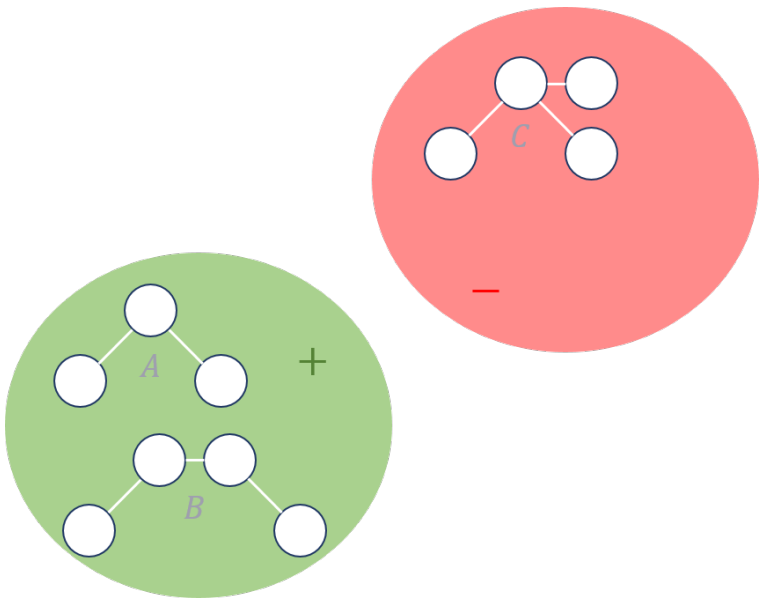
- Limit the relative weight updates by setting a **ceiling**
- Switch to go from relative to **absolute** push/pull increments
- Other, cleaned datasets. And **OGB** datasets
- Split the dataset in training and test data.
  And evaluate for **overfitting**. Run for many epochs and search for some kind of plateau in the learning.

Thank you all for listening.

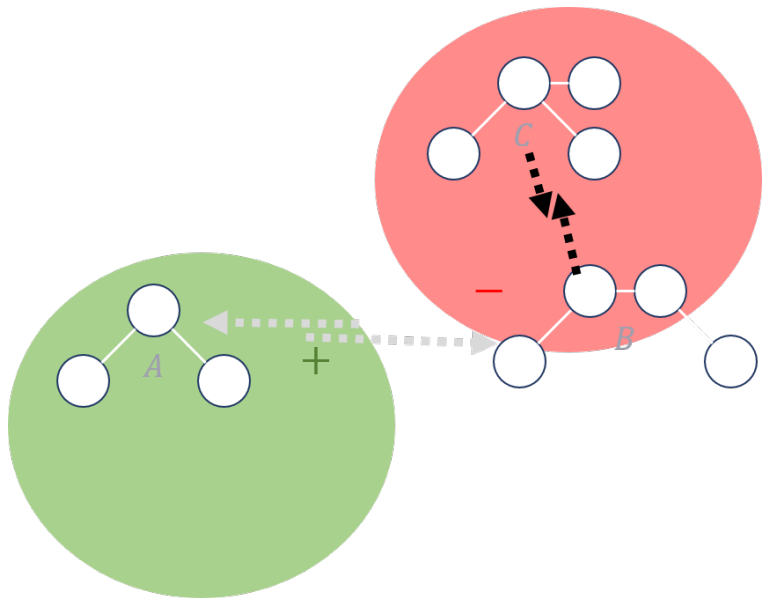
I will be happy to answer any **questions** and hear your **comments**.

Example of the whole procedure

Current clustering:



Target clustering:



Idea: Reduce distance between B and C, by updating the edge weights.

Preparation of the performance comparison

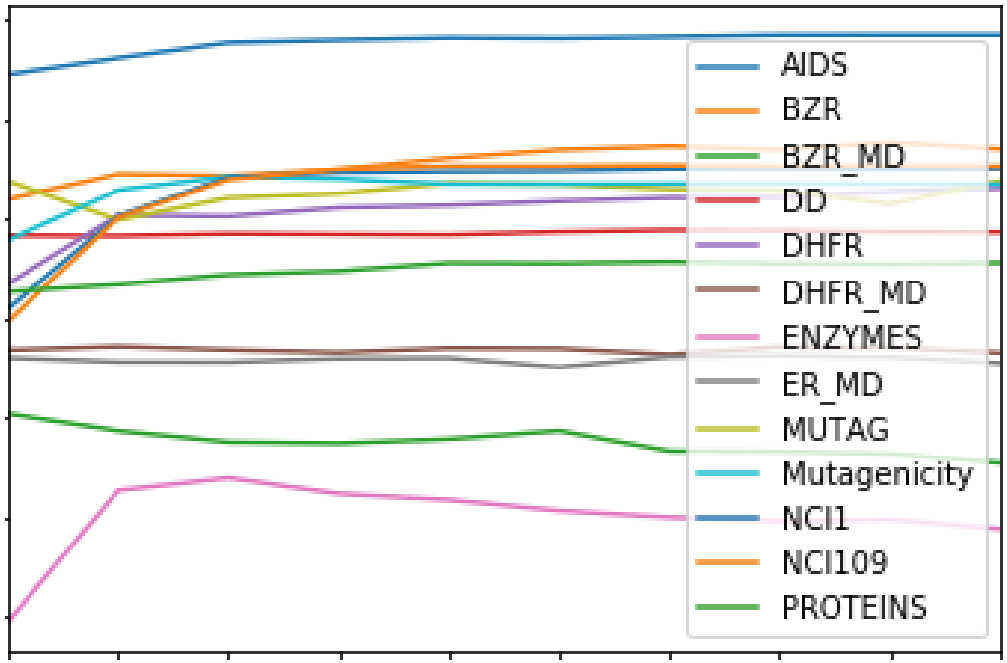


Figure: Classification accuracies on databases using Weisfeiler-Lehman.

```
grakel.kernels.WeisfeilerLehman(n_iter=[1-10], base=grakel.kernels.VertexHistogram, normalize=True)
grakel.utils.cross_validate_Kfold_SVM(K, y, n_iter=10)
```

Implementation road-map 1/2

UNIVERSITÄT BONN ML AI lab

- **WLLT Construction**:
  - Write to file and read from file. Construct WL-iteration based.
  - All weights *equal*.
  - (*Random* initial weights.)
  - (Use *a priori* knowledge.)

- **Wasserstein-Distance feedback**:
  - "Biggest pile of dirt". ("Smallest", to increase the distance.)
  - Distribution proportional to the pile size.
  - Distribution proportional to the cost of moving the pile size.

Implementation road-map 2/2

- **Update rule**:
  - Value:
    - Constant λ.
    - *Gradient descent*.
  - Location:
    - *Local*: Only update the first and last edge weights of the connecting path.
    - *Weighted path*: Update all edge weights on the path, with less magnitude for edges closer to the root.
    - *Path*: Update all edges on the path.
    - *Global*: Update all edges, related to all occurring labels.