Towards Decentralized GNNs

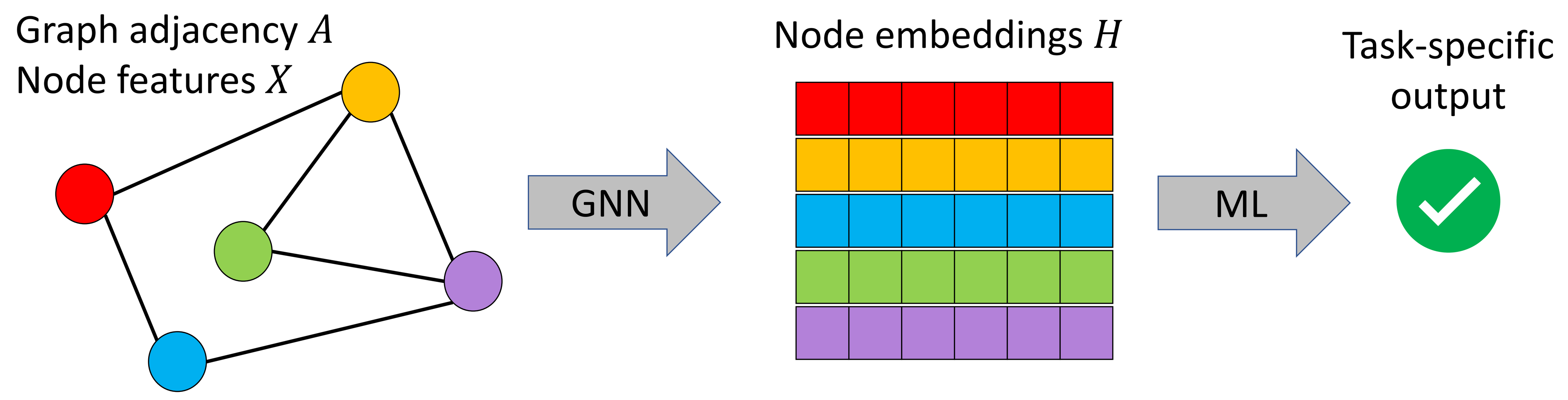
Lodovico Giaretta
lodovico@kth.se

Sarunas Girdzijauskas
sarunasg@kth.se

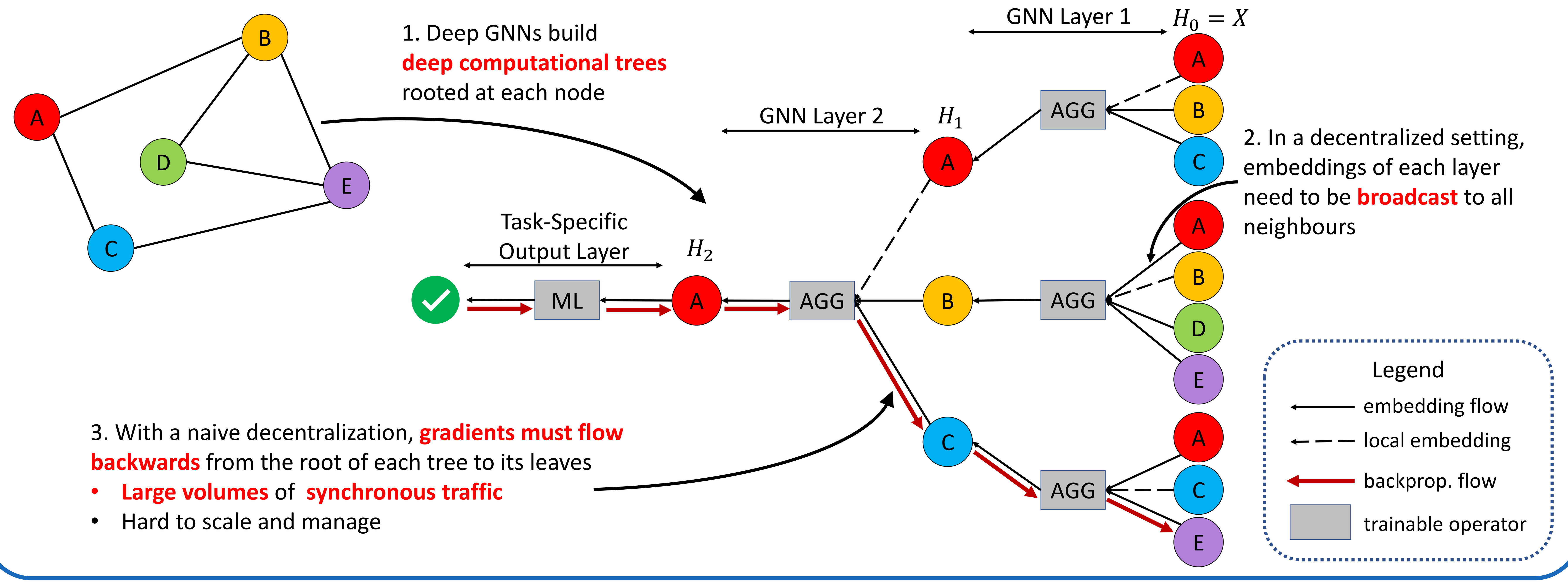
Setting: Decentralized GNNs for Device Embedding

Scenario: a **decentralized network** of interconnected nodes (e.g. IoT devices)

Objective: use **Graph Neural Networks** (GNNs) to build **node embeddings** to perform one or more tasks (e.g. device classification)



The Problem: Decentralizing GNN Gradients



1. Deep GNNs build **deep computational trees** rooted at each node

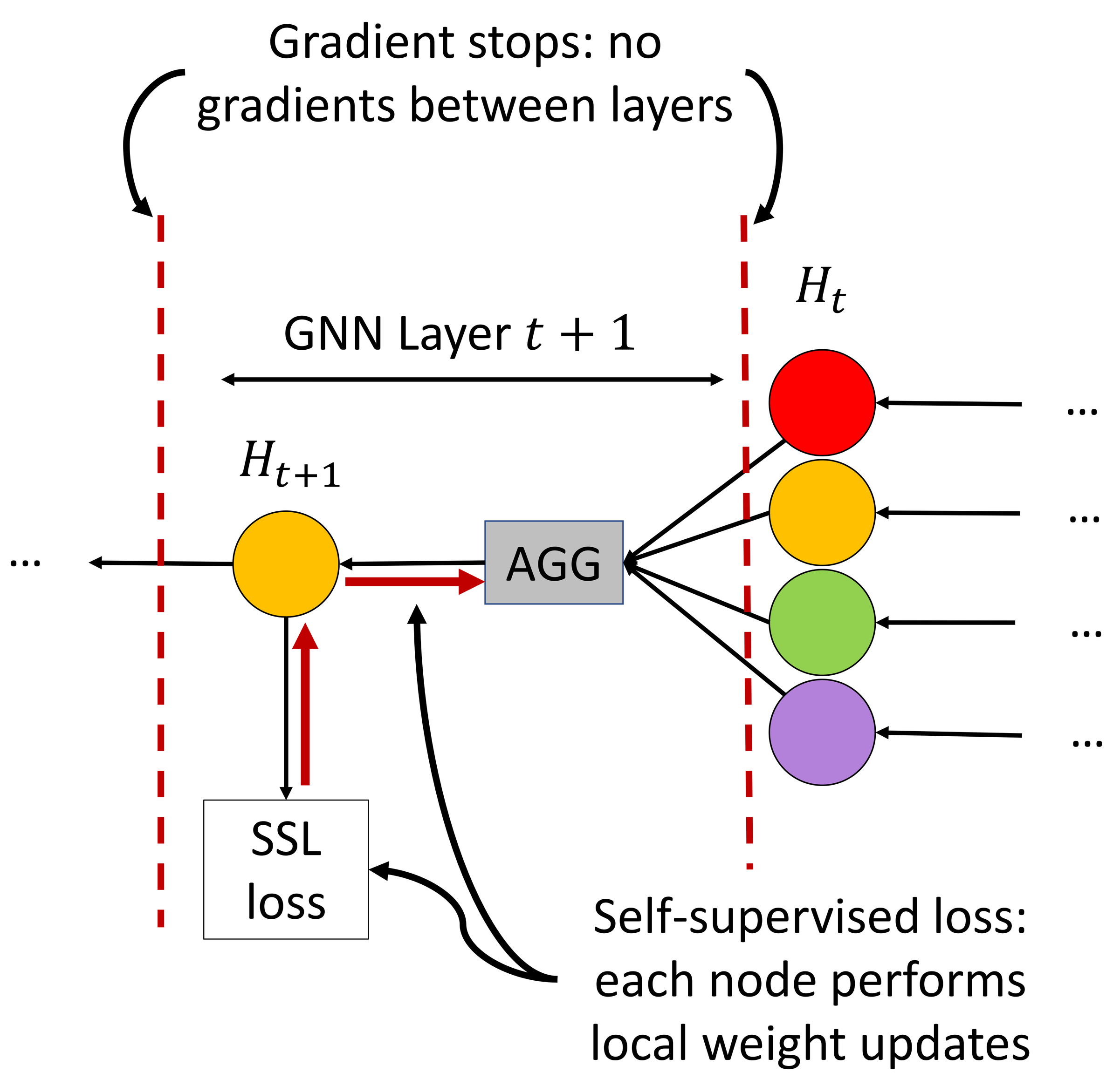
2. In a decentralized setting, embeddings of each layer need to be **broadcast** to all neighbours

3. With a naive decentralization, **gradients must flow backwards** from the root of each tree to its leaves
- **Large volumes** of **synchronous traffic**
- Hard to scale and manage

The Solution: Layer-Wise Self-Supervision

Layer-wise Training: each layer treats its inputs as **constants**
- Do **not send gradients** to previous layers
- Do **not receive gradients** from next layers
- Each node learns the best output locally, in an **isolated** way
- Input embeddings are stored locally until a new version is received, enabling **asynchronous training**

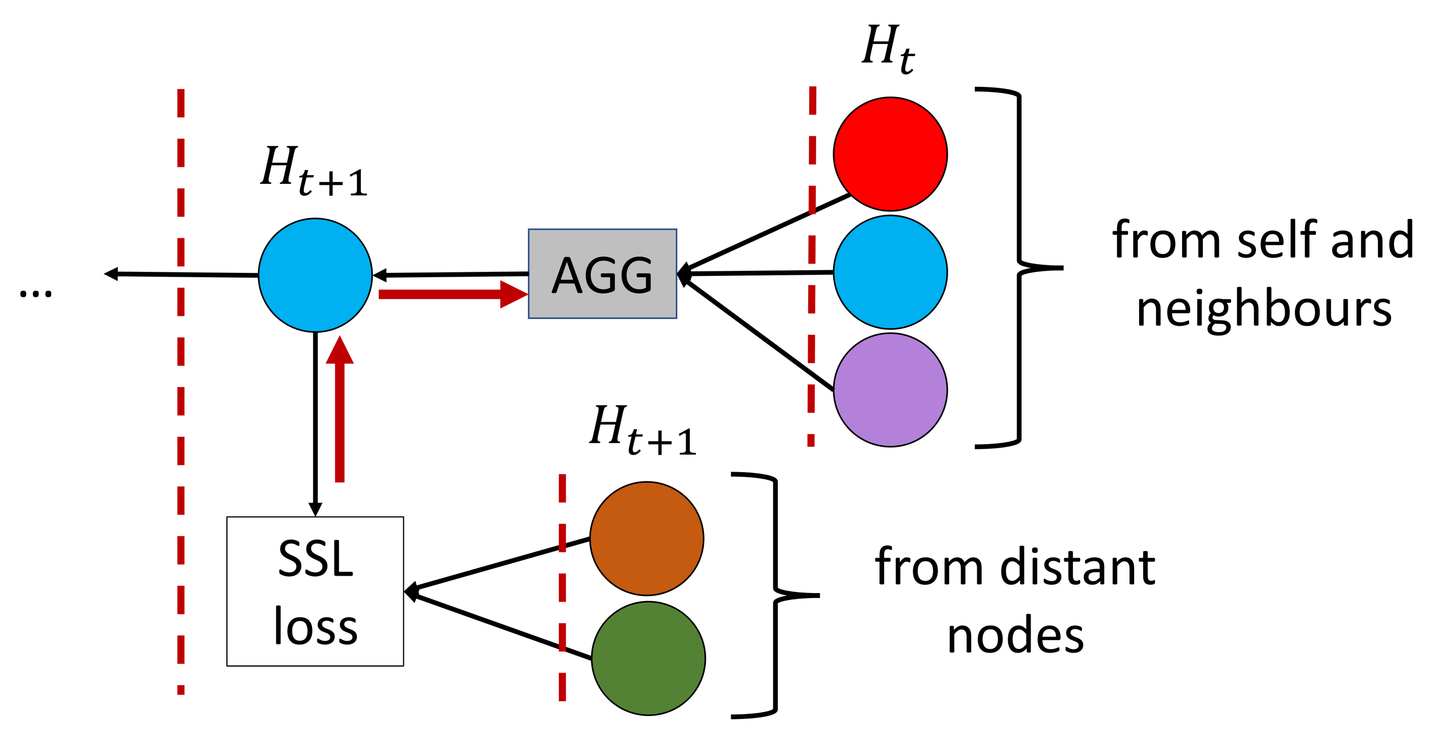
Self-Supervised Learning (SSL): employ loss functions based on **implicit input information**, without any labels
- Extract **all relevant phenomena** in the input
- Learn high-quality, **task-agnostic** embeddings



Negative Sampling

Problem: Self-supervised learning requires **negative samples** in one of two ways:
- **Explicitly** in **contrastive** SSL (e.g. edge reconstruction loss)
- **Implicitly** in **non-contrastive** SSL (e.g. batch-wise orthogonality constraints)

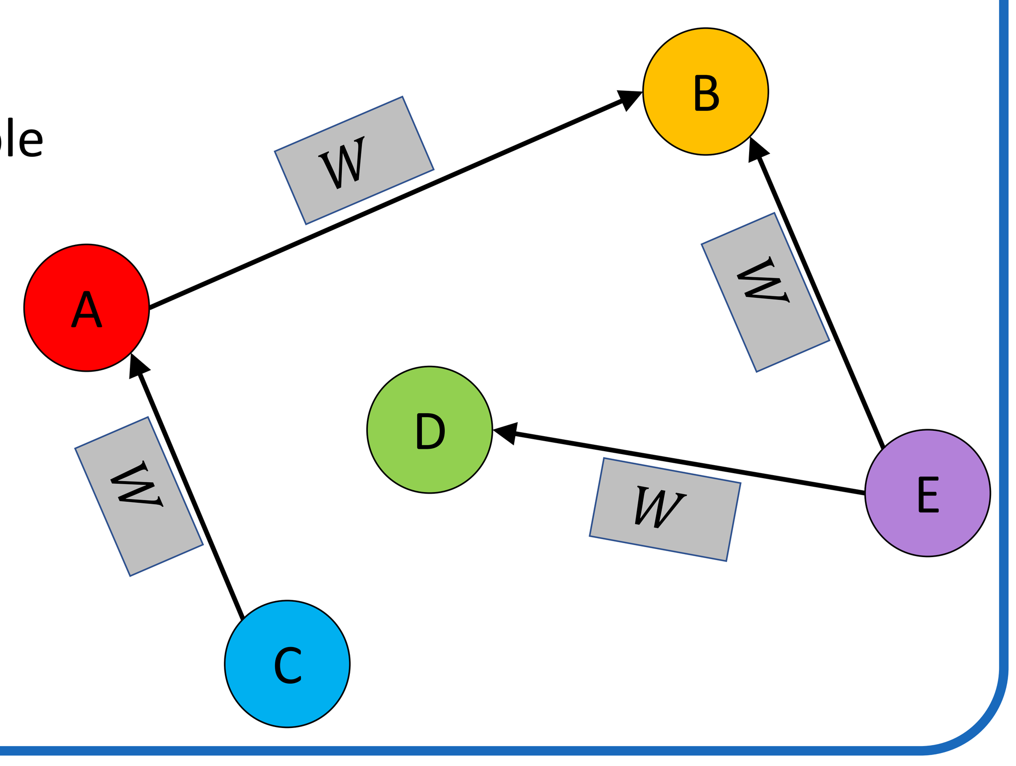
Solution: Push-based, asynchronous random sampling of embeddings
- Each node **periodically shares** its embedding with a few randomly-picked nodes



Gossip Learning

Problem: need to train a **global** GNN model
- Each node is training the model parameters **independently**
- These parameters need to be **shared and merged** to reach convergence

Solution: Gossip Learning
- Each node **periodically sends** its trainable parameters to a randomly-picked node
- The receiver **merges** the received parameters and its local ones, then performs additional **local training**



MARIE CURIE ACTIONS

Work funded by EU's Horizon 2020 program under Marie Curie grant agreement ITN – RAIS No 813162
https://rais-itn.eu