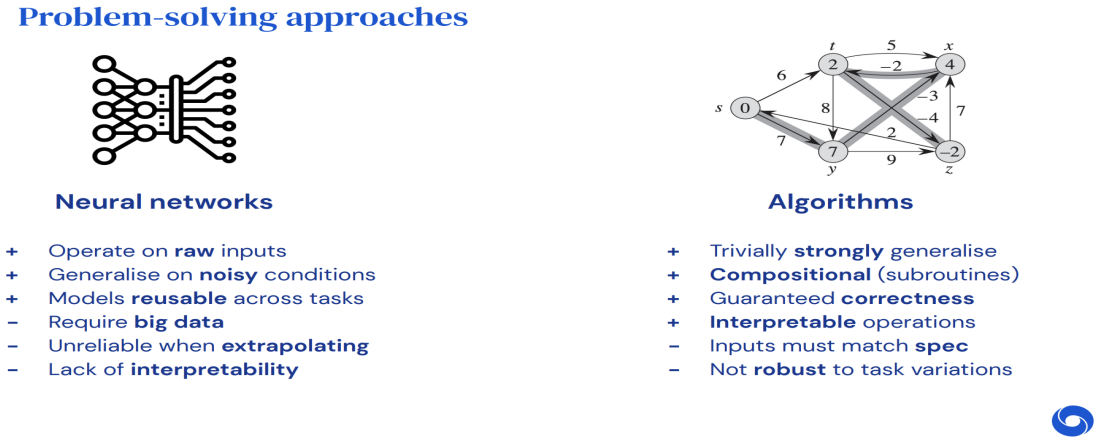
An algorithmic reasoning approach to GNNs

Graphs are an important building block since they can naturally encode an entity-relationship structure, as well as an invariance to permutations (of both nodes and edges) and awareness of input sparsity. The reasoning we do on a problem represented by a graph strongly resembles the characteristics that define computational thinking: decomposition, pattern recognition / data representation, generalization/abstraction, and algorithms.



GNN can have a lot of meanings, there isn't just one architecture that can be recognized as "GNN". We will try to understand the general, abstract structure of a GNN that is presented in the paper *Relational inductive biases, deep learning, and graph networks*, Battaglia et al. (link) and to shed light about the relational inductive bias and combinatorial generalization of a GNN.

Our motivation is to better understand the extent to which graph neural networks are capable of precise and logical reasoning. In respect to NP-hard problem solving, there have been many problem approaches designed to implement GNNs, but all of these approaches aim to assist an "hand made" solver like in the paper *Combinatorial Optimization and Reasoning with Graph Neural Networks*, Morris et al. (link) We want to deploy a GNN that solves a problem by itself.

An interesting but by far more complex task that can be achieved end-to-end by a GNN is the SAT problem, presented in the paper *Learning a SAT solver from single-bit supervision*, Selsam et al. (link). The SAT problem is NP-Hard, hence every NP problem can be reduced to SAT in polynomial time, so solving the former with a GNN would solve them all!! (super cool)

Then we will train a GNN to solve a sorting problem. Why a sorting problem?

- Infinite training data
- No noise in the training data
- Complex data manipulation

Bibliography:

- *Relational inductive biases, deep learning, and graph networks*, Battaglia et al. (link)
- *Combinatorial Optimization and Reasoning with Graph Neural Networks*, Morris et al. (link)
- *Learning a SAT solver from single-bit supervision*, Selsam et al. (link)