ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE

TrUstworthy Planning and scheduling with Learning and ExplanationS

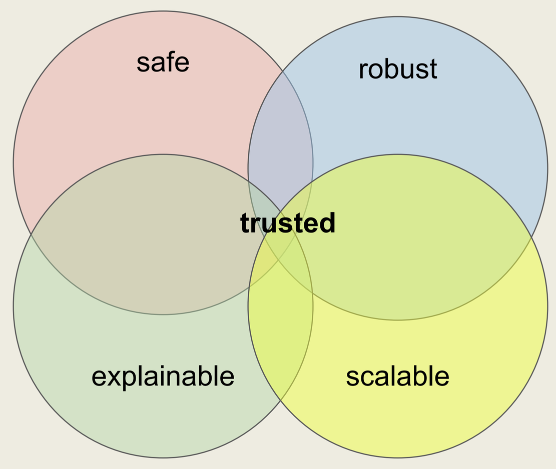
Tuples TRUSTWORTHY AI

Project Overview

Objective: Build Trustworthy Planning & Scheduling Systems

Planning & Scheduling: key problem in AI and autonomous systems
- Decision making over time (choice of actions and their timing)
- Objectives (e.g. achieving a goal, minimizing a cost)
- Constraints (e.g. time, resources)

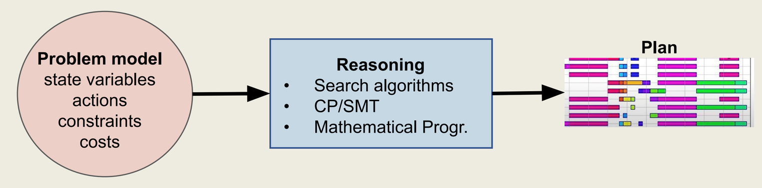
Widespread application in industrial and governance sectors



Horizon Europe — Funded by the European Union
- Research and Innovation Action
- Oct 2022 - Sept 2025

Existing Approaches Are Not Trustworthy

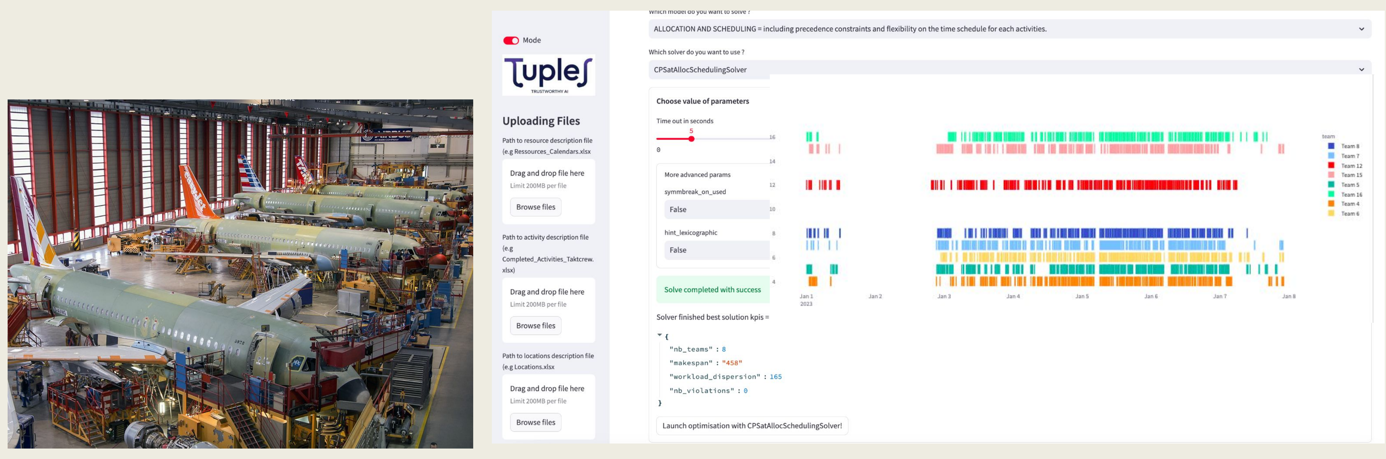
- Model-based (reasoning) methods
  safe ✔ explainable ✔ stable ✔ adaptable ☐ scalable ☐



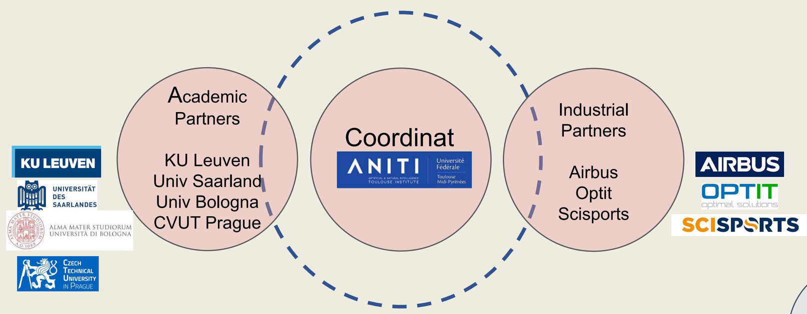
- Data-driven (learning) methods
  safe ☐ explainable ☐ stable ☐ adaptable ✔ scalable ✔

Example: Aircraft Manufacturing

AIRBUS



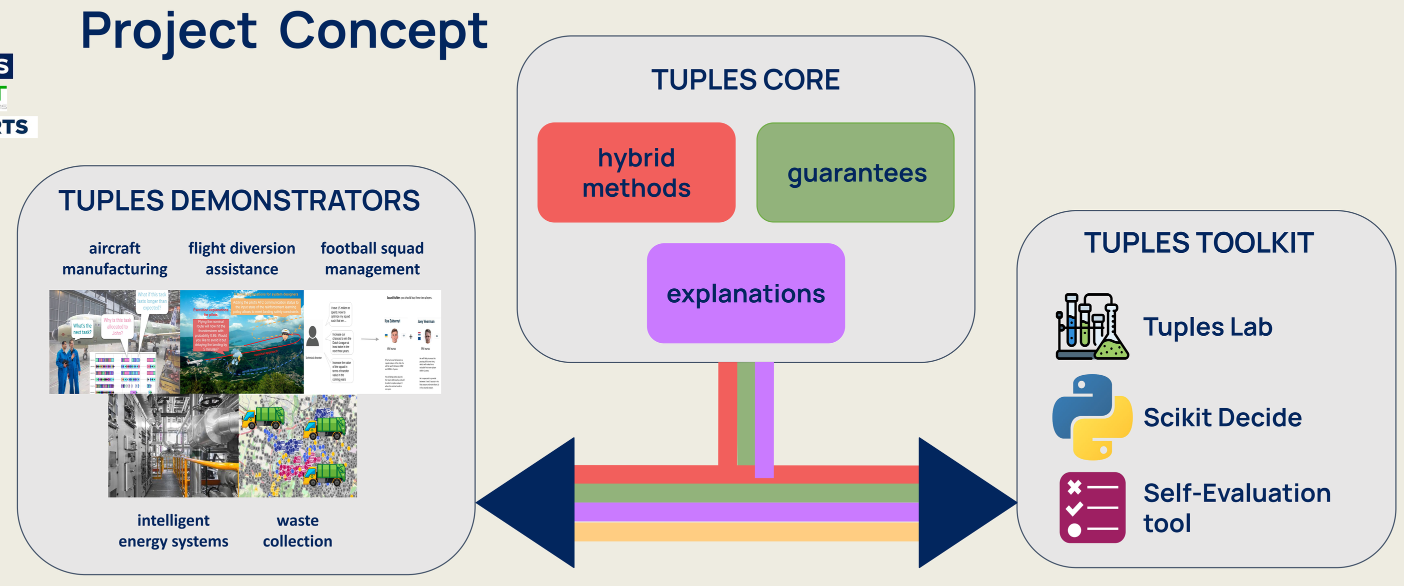
- Thousands of activities to schedule and assign to resources
- Constraints: precedence, workforce and machine availability, skills...
- Objectives: costs, completion time, quality...
- Many sources of uncertainty: delays, equipment failures ...



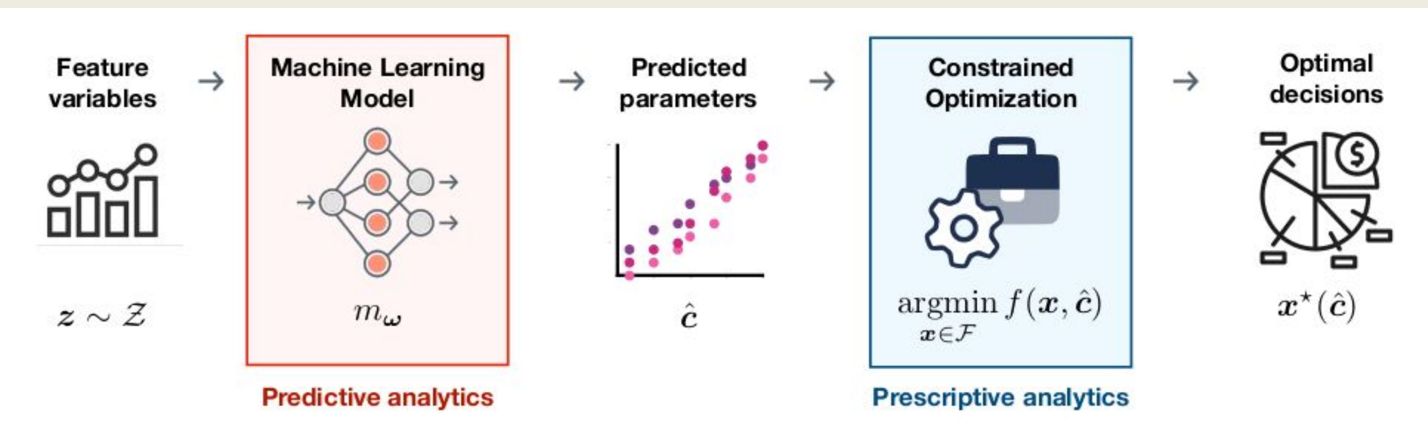
Hybrid Methods

Three main lines of work:
- Learning the problem's constraints
- Learning the problem's objective – Decision-focused learning
- Learning policies and heuristics to guide the search for a solution

Project Concept



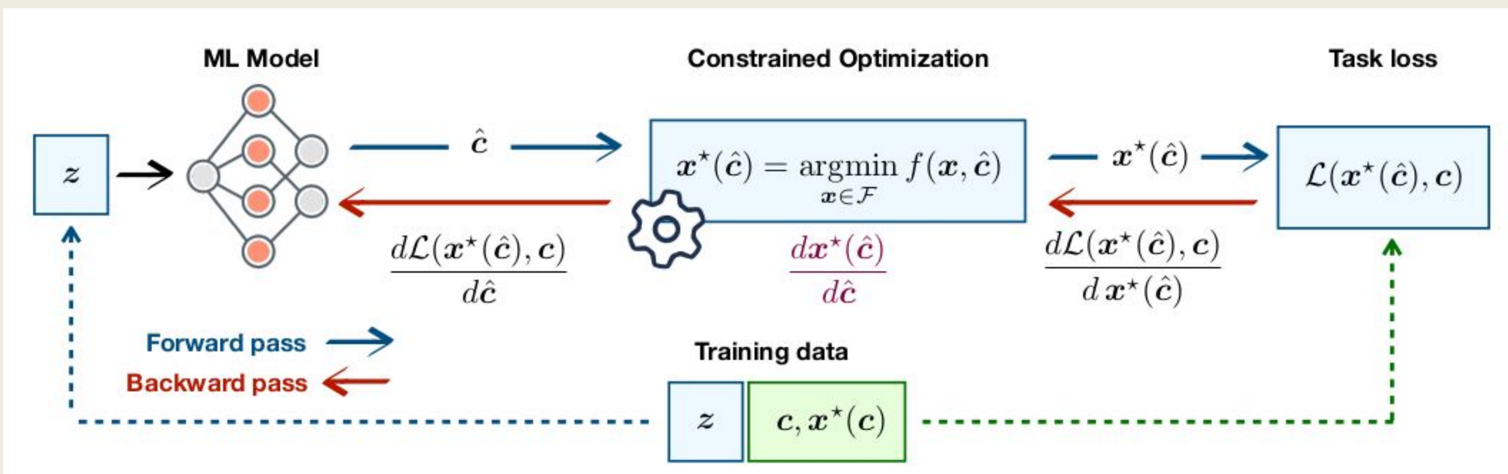
Highlight 1: Decision-focused learning



Setting = Prediction + Optimisation

New family of methods: decision-focused learning

- Challenge 1: **Non-differentiable** constraint solver
- Challenge 2: **Computationally intensive** constraint solving



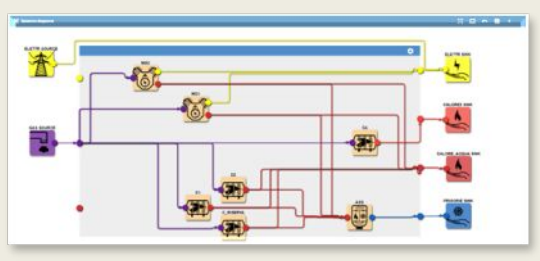
DFL survey [Mandi et al, '24]

	Objective Function	Constraint Functions	Decision Variables	CO Solver
Analytical Differentiation of Optimization Mappings	Strictly Convex	Convex	Continuous	Primal-Dual Solver
Analytical Smoothing of Optimization Mappings	Linear	Linear	Continuous/ Discrete	Primal-Dual Solver
Smoothing by Random Perturbations	Linear in the Predicted Parameter	Not limited to specific form	Continuous/ Discrete	Solver agnostic
Differentiation of Surrogate Loss Functions	Linear in the Predicted Parameter	Not limited to specific form	Continuous/ Discrete	Solver agnostic

Utilize

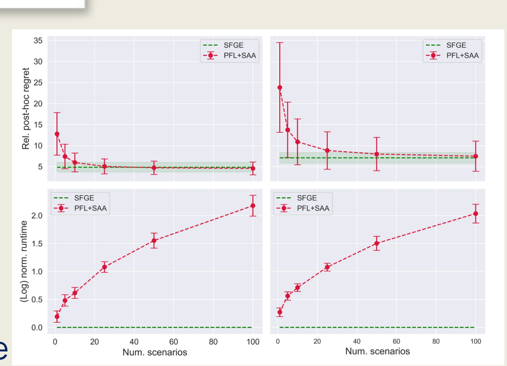
https://github.com/PredOpt/predopt-benchmarks

Example use case: Energy Production Management Under Uncertainty



Given:
- Energy supply & demand
- Price and demand predictions
- Information on how uncertainty might unfold

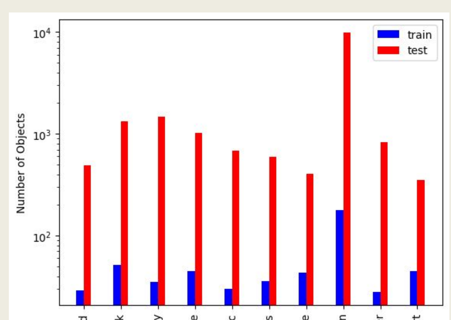
Compute:
- Routing/generation decisions over time with minimal cost
- Satisfying all demands
- Robust w.r.t. uncertainty



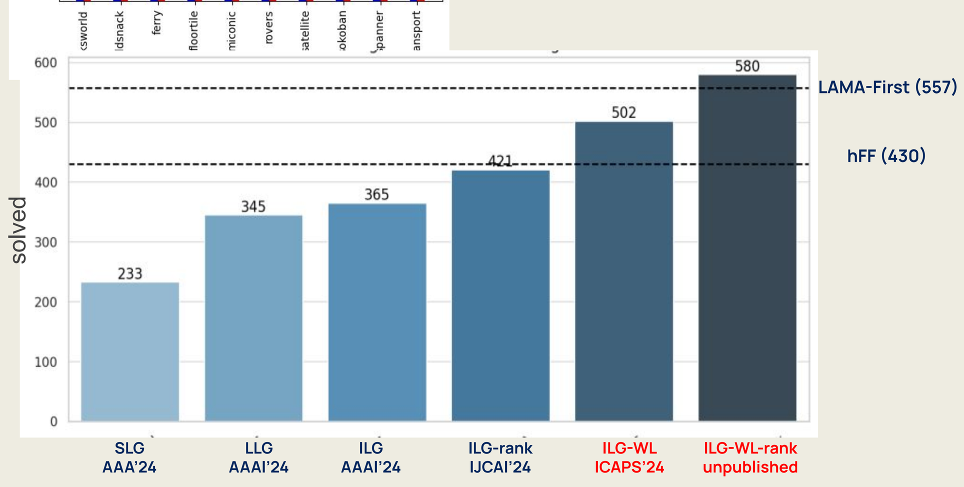
Result: Comparable quality w.r.t. Monte-Carlo approximate methods but ~100x faster at inference time

Results

International Planning Competition learning track

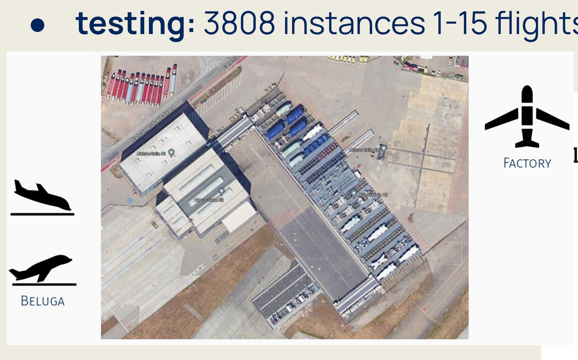


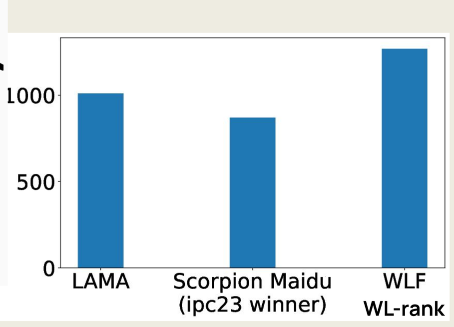
- 10 domains
- **training:** 99 small instances
- **testing:** 3x30 easy/med/large



Beluga Logistics Planning (Airbus)

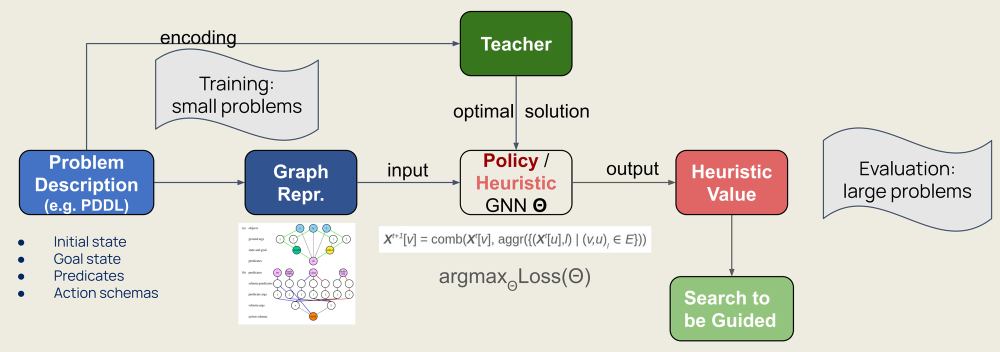
- **training:** 308 instances 1-3 flights, 2-23 parts, 2-8 racks
- **testing:** 3808 instances 1-15 flights, 3-64 parts, 2-19 racks



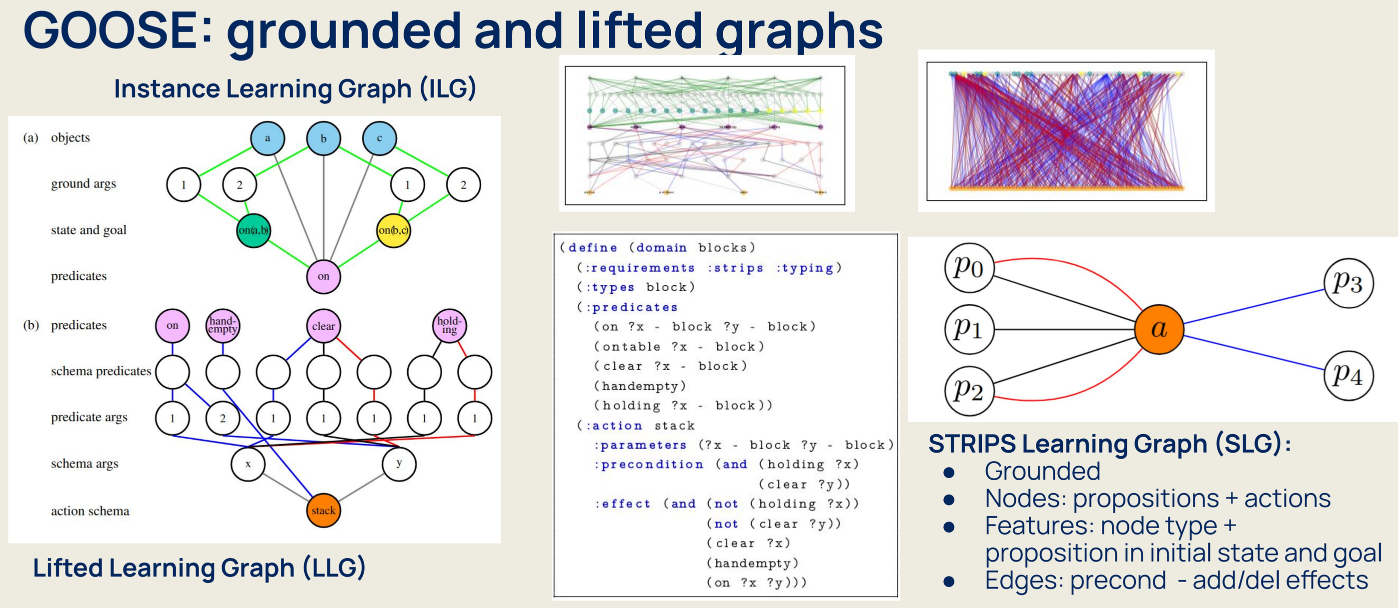


Highlight 2: Learning planning heuristics

Supervised learning of **heuristics** to guide search
- architectures exploit the structure of **symbolic** planning representations
  ⇒ data **frugal**, **generalises** to larger problems, **robust** to domain changes
- GOOSE family of planners

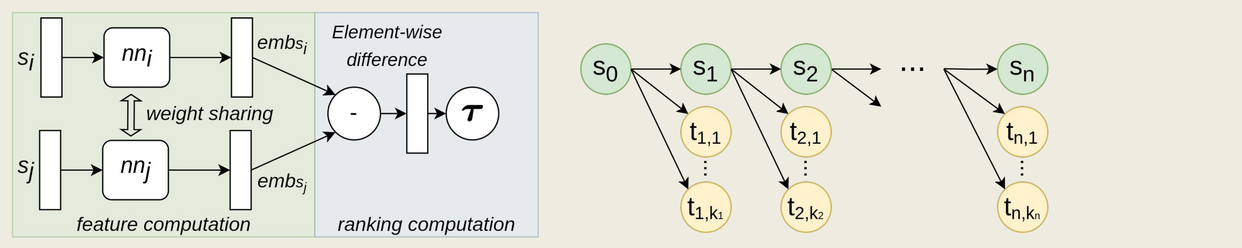


GOOSE: grounded and lifted graphs



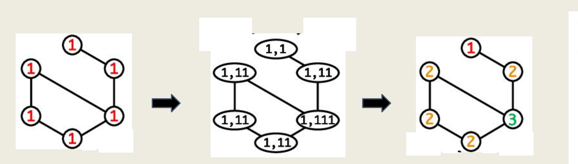
GOOSE-Rank: forget heuristics - learn to rank states!

- Greedy Best First Search: **ranking states is sufficient** and more flexible
- **optimal ranking:** rank states on optimal path as better than those off it
- **pairwise direct ranking** enforces transitivity and gets more data for free!
- combines with and improves a NN regression model (e.g GOOSE GNNs)



WL-GOOSE: forget GNNs, use classical ML!

Key issues with GNNs
- **slow** at inference time
- **expressive power limited** by Weisfeiler-Leman (WL) algorithm
- incomplete test for graph non-isomorphism
- Indistinguishable states: GNNs **cannot learn** certain heuristics



Algorithm 1: WL algorithm
1. $c^0(v) \leftarrow c(v), \forall v \in V$
2. **for** $j = 1, \ldots, h$ **do for** $v \in V$ **do**
3. $c^j(v) \leftarrow \text{hash}\,(c^{j-1}(v), \{\!\{c^{j-1}(u) \mid u \in \mathcal{N}(v)\}\!\})$
4. **return** $\bigcup_{j=0,\ldots,h} \{\!\{c^j(v) \mid v \in V\}\!\}$

Alternative: classical ML + graph features
- **features:** frequency of colors produced by the WL algorithm
- can use any **classical ML technique** (SVM, GP)
- same expressiveness as GNNs
- **100-1000x faster** training and inference

ANITI Université de Toulouse

Point of contact (project lead): Sylvie THIEBAUX, sylvie.thiebaux@laas.fr

Don't miss our challenge!