MACQ

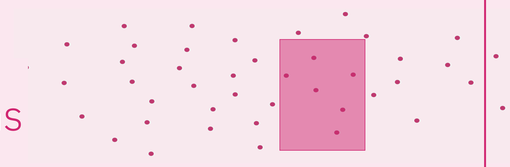
https://macq.planning.domains

Model Acquisition Toolkit

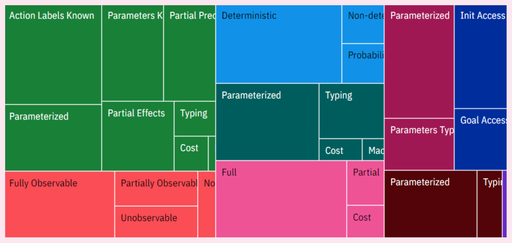
Ethan Callanan
Rebecca De Venezia
Victoria Armstrong
Alison Paredes
Jungkoo Kang
Tathagata Chakraborti
Christian Muise

Contact christian.muise@queensu.ca

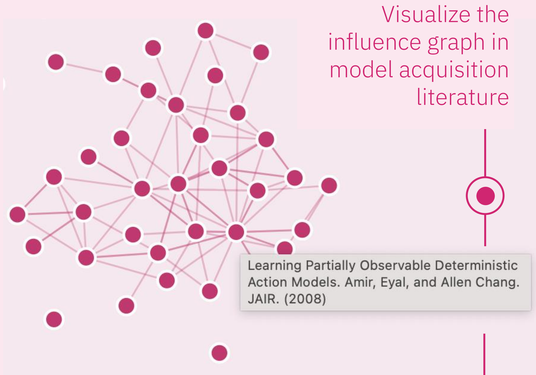
Model acquisition literature seen through document embedding



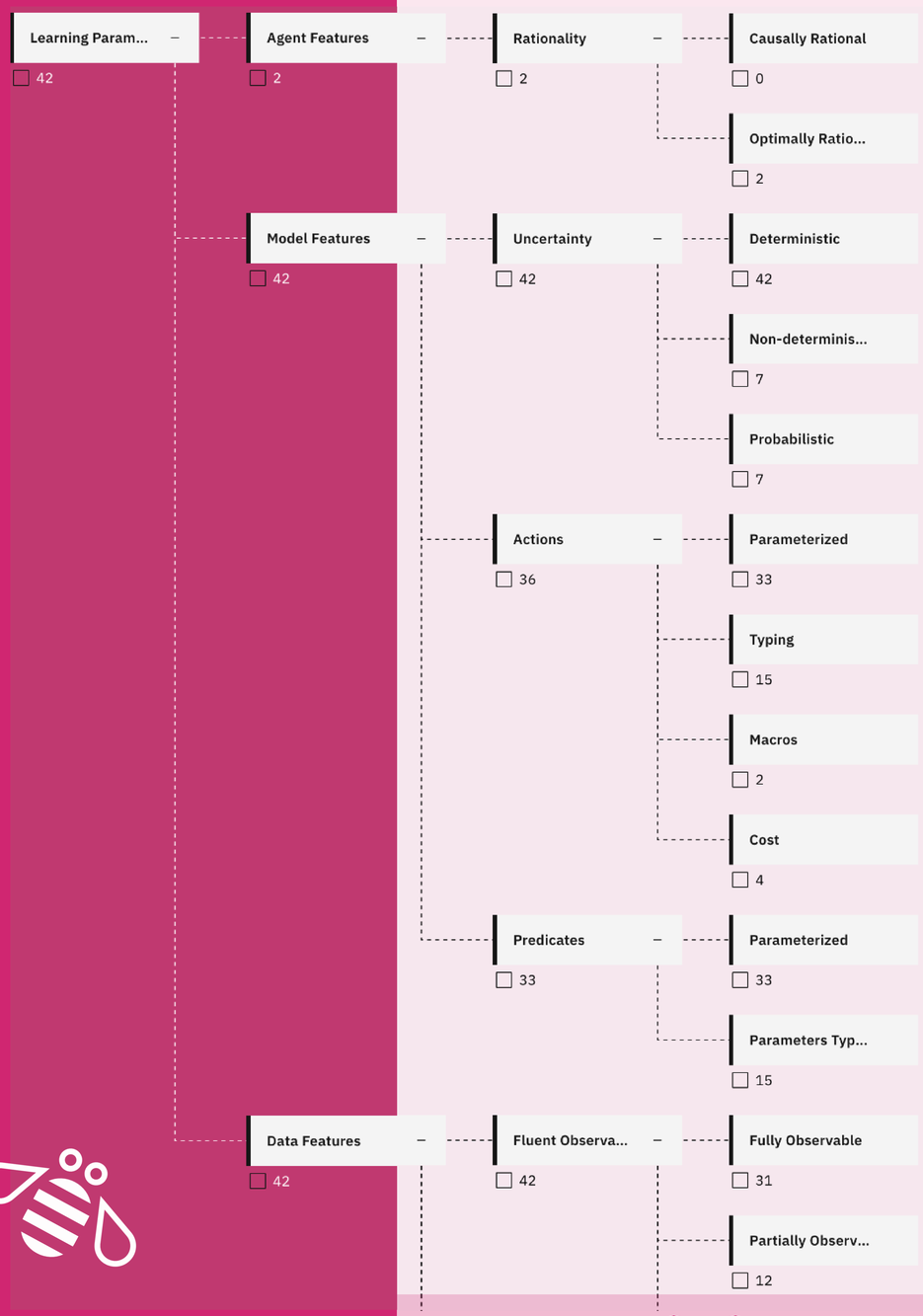
Explore Treemap of concepts in model acquisition literature



Visualize the influence graph in model acquisition literature



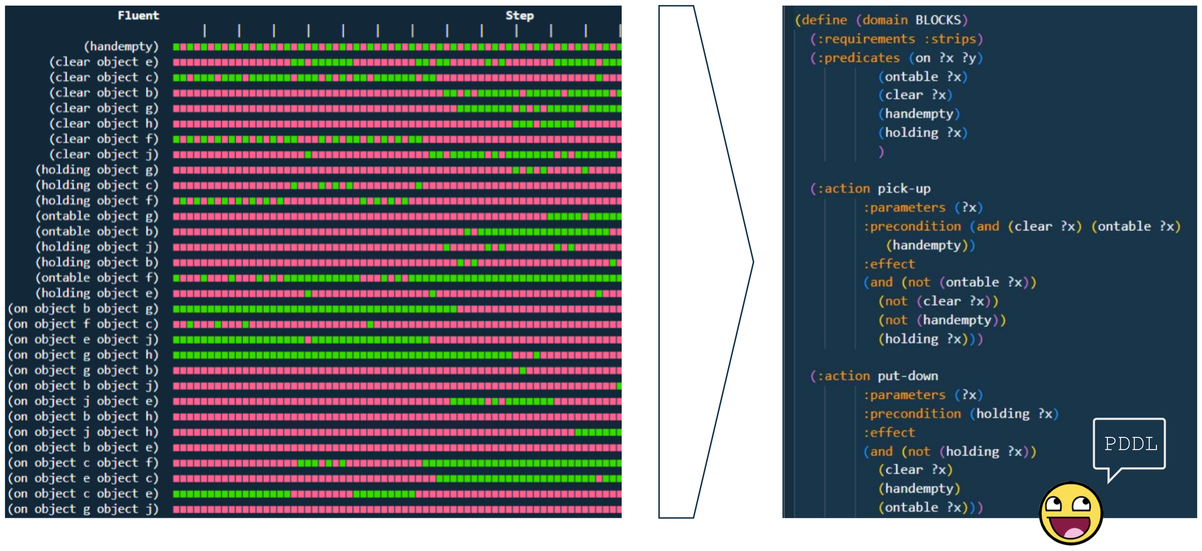
Climb the hierarchy of concepts



For over three decades, the planning community has explored countless methods for data-driven model acquisition. MACQ aims to be the one-stop shop for model learning techniques.

We have started of with re-implementations of landmark approaches in the field. MACQ is open source, and we welcome contributions! 🤗

Observed Behavior → Action Model




```
(define (domain BLOCKS)
  (:requirements :strips)
  (:predicates (on ?x ?y)
               (ontable ?x)
               (clear ?x)
               (handempty)
               (holding ?x)
               )

  (:action pick-up
           :parameters (?x)
           :precondition (and (clear ?x) (ontable ?x)
             (handempty))
           :effect
           (and (not (ontable ?x))
             (not (clear ?x))
             (not (handempty))
             (holding ?x)))

  (:action put-down
           :parameters (?x)
           :precondition (holding ?x)
           :effect
           (and (not (holding ?x))
             (clear ?x)
             (handempty)
             (ontable ?x)))
```

PDDL

MACQ API

```python
from macq import generate, extract
from macq.trace import PlanningObject, Fluent, TraceList
from macq.observation import PartialObservation

def get_fluent(name: str, objs: list[str]):
    objects = [PlanningObject(o.split0[0], o.split()[1])
               for o in objs]
    return Fluent(name, objects)

traces = TraceList()
generator = generate.pddl.TraceFromGoal(problem_id=1801)
for _ in range(100):
    traces.append(generator.generate_trace())
```

MACQ comes with a customizable out-of-the-box trace generator

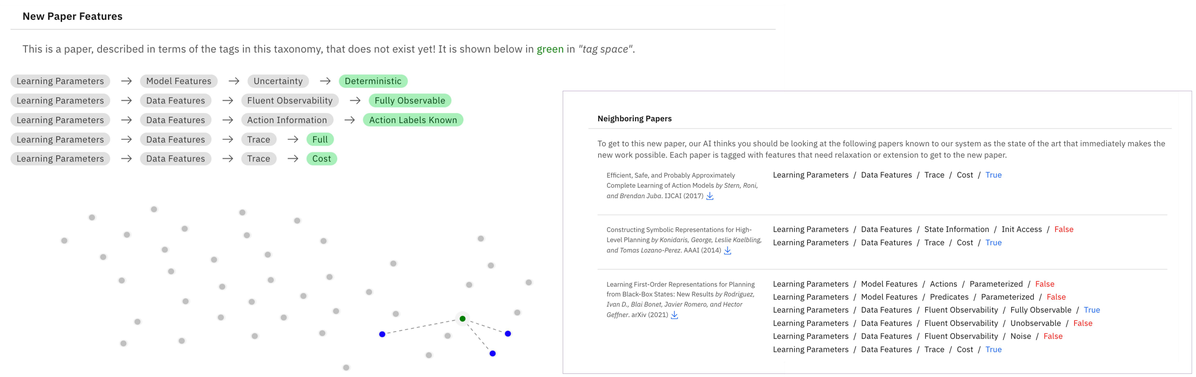
Our tokenization feature allows users of MACQ to transform data in terms of features like noise and observability.

```python
observations = traces.tokenize(
    PartialObservation,
    percent_missing = 0.60
)
model = extract.Extract(
    observations,
    extract.modes.ARMS,
    upper_bound = 2,
    min_support = 2,
    action_weight = 110,
    info_weight = 100,
    threshold = 0.6,
    info3_default = 30,
    plan_default = 30,
)
print(model.details())
```

Extraction API accepts parameterized requests for model acquisition.

What's Next

Our holistic characterization of action model acquisition literature offers deep insight into the research opportunities that remain in terms of papers, in the space of our taxonomic classes, that have not been written yet!



This is a sparkline of action model acquisition research from 1989 to 2022. It's time for you to join the fun! 🤓

https://github.com/AI-Planning/macq

https://pypi.org/project/macq

CONTRIBUTE