OAK RIDGE National Laboratory

Dynamic Resource Allocation for Reasoning Agents in HPC Workflows

Ana Gainaru

Denise Adorno Lopes
Guillaume Pallez

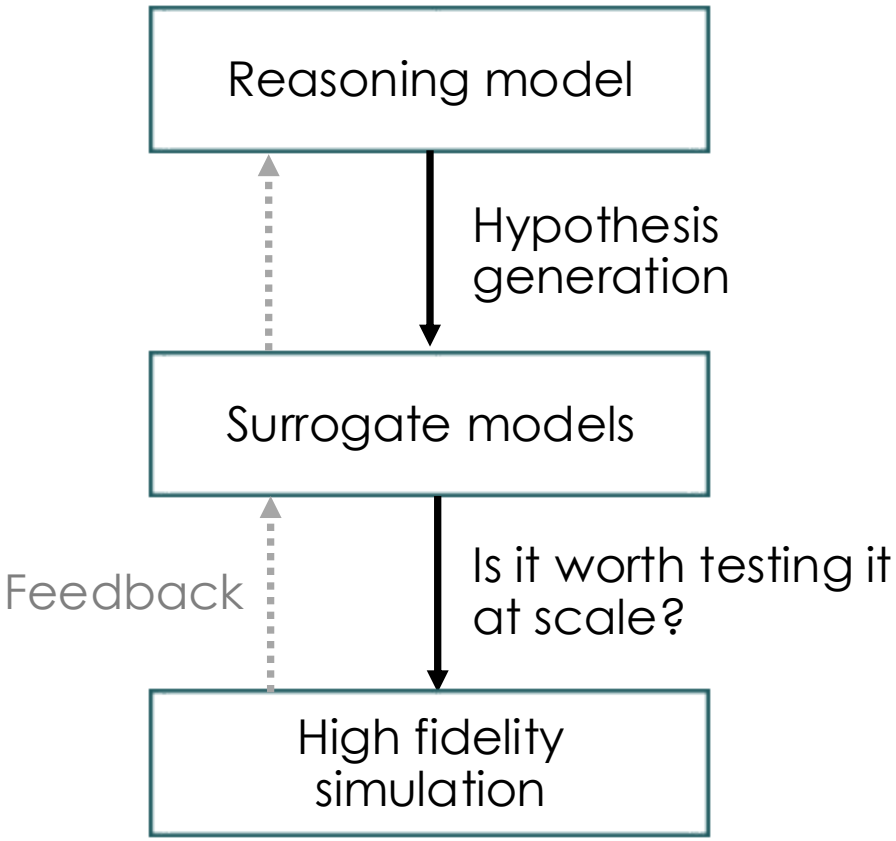
19th workshop on Scheduling for large-scale systems. Frejus, France, March 16-19, 2026

ORNL is managed by UT-Battelle LLC for the US Department of Energy

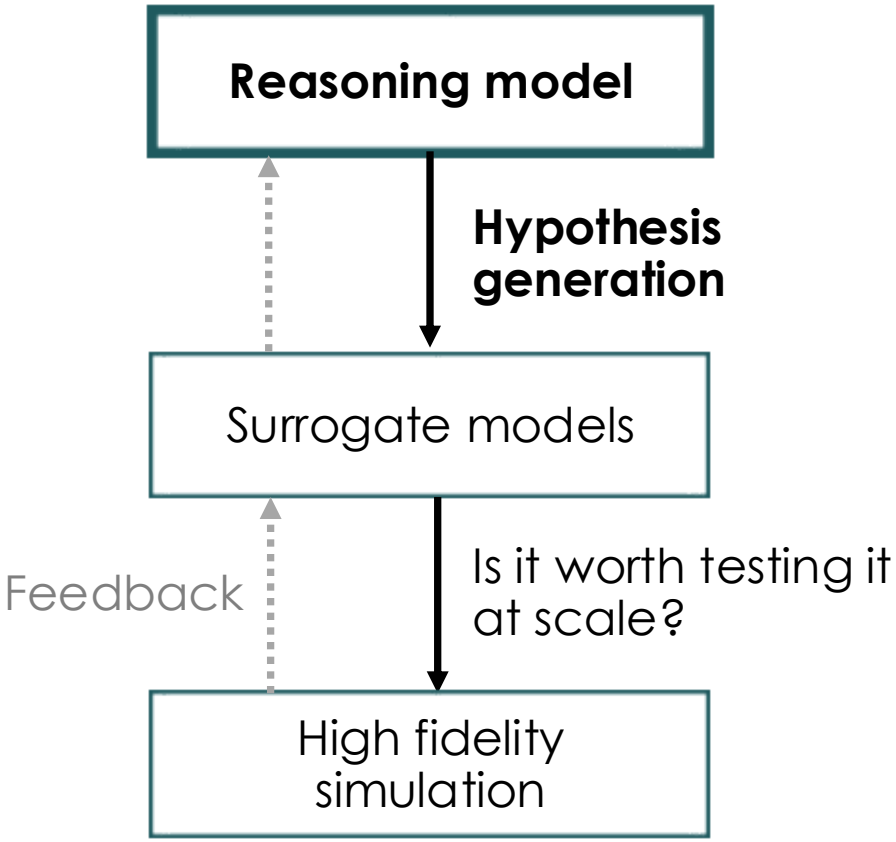
U.S. DEPARTMENT OF ENERGY

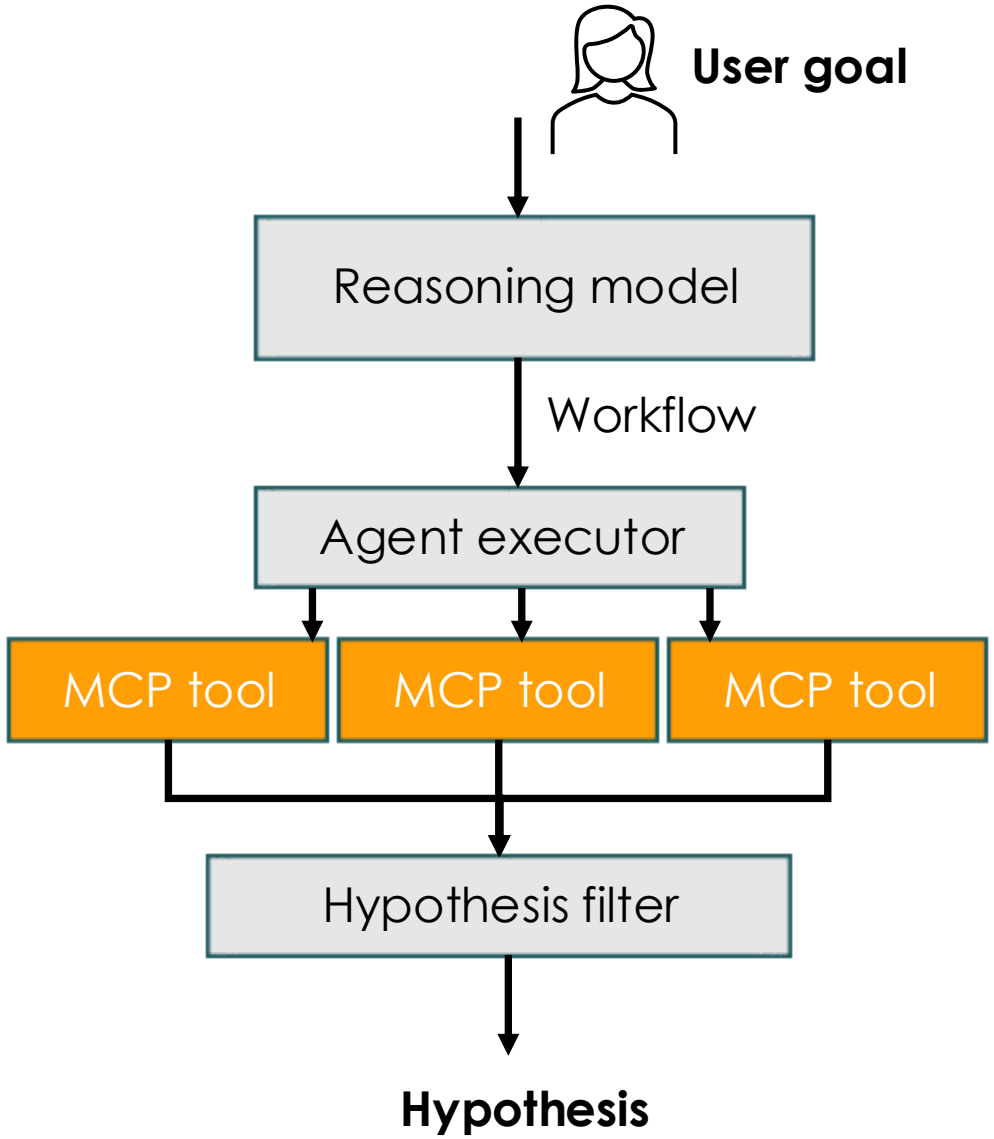
gainarua@ornl.gov

AI applications using reasoning models

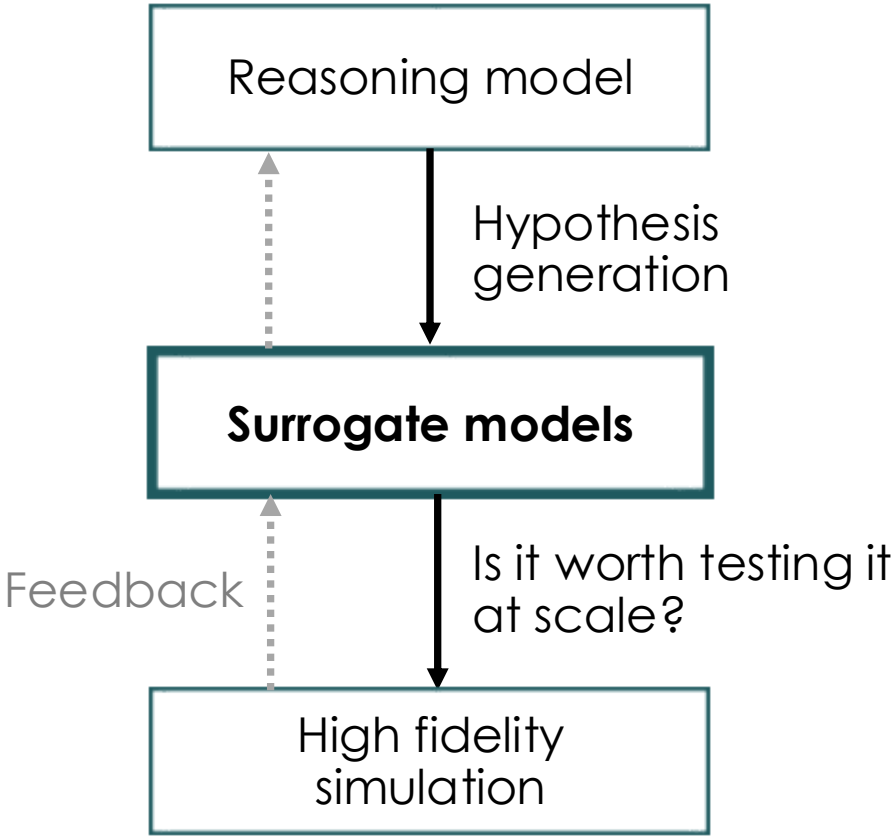


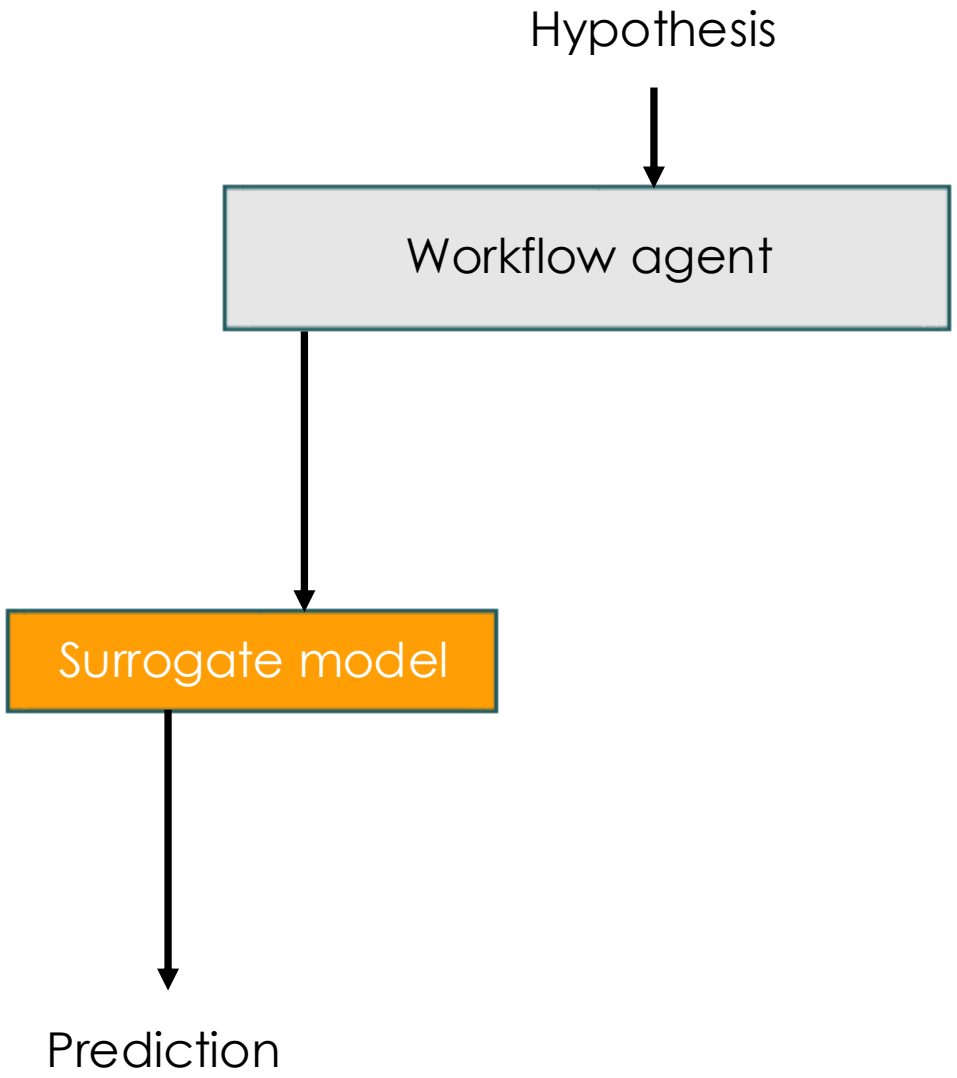
AI applications using reasoning models



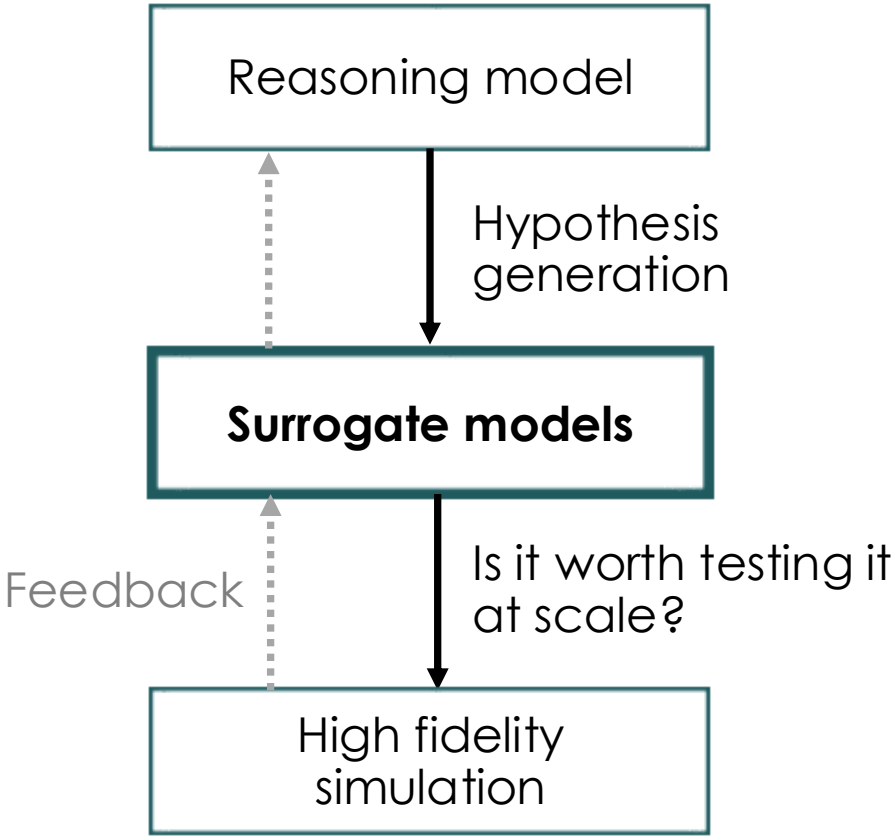


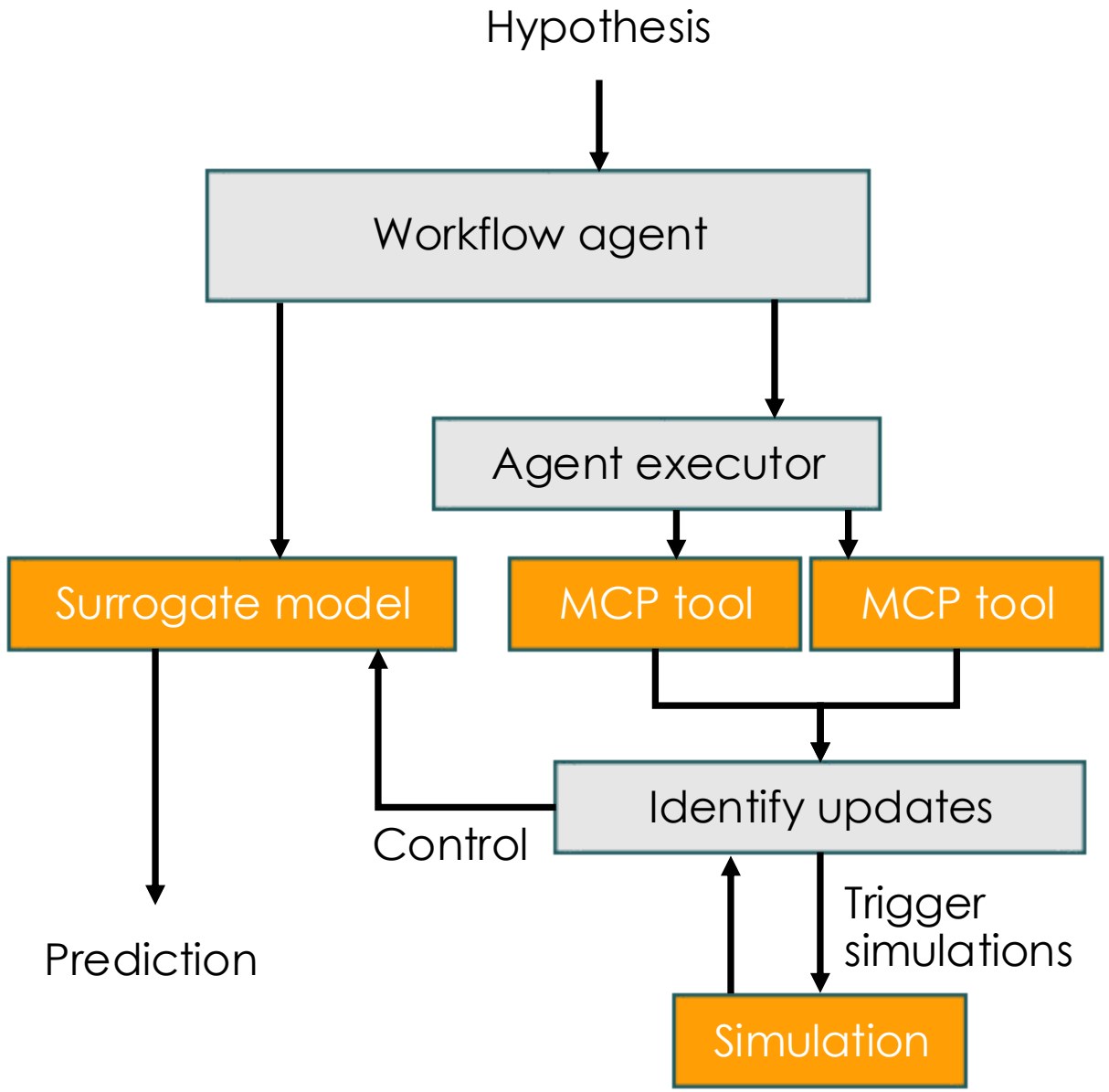
AI applications using reasoning models





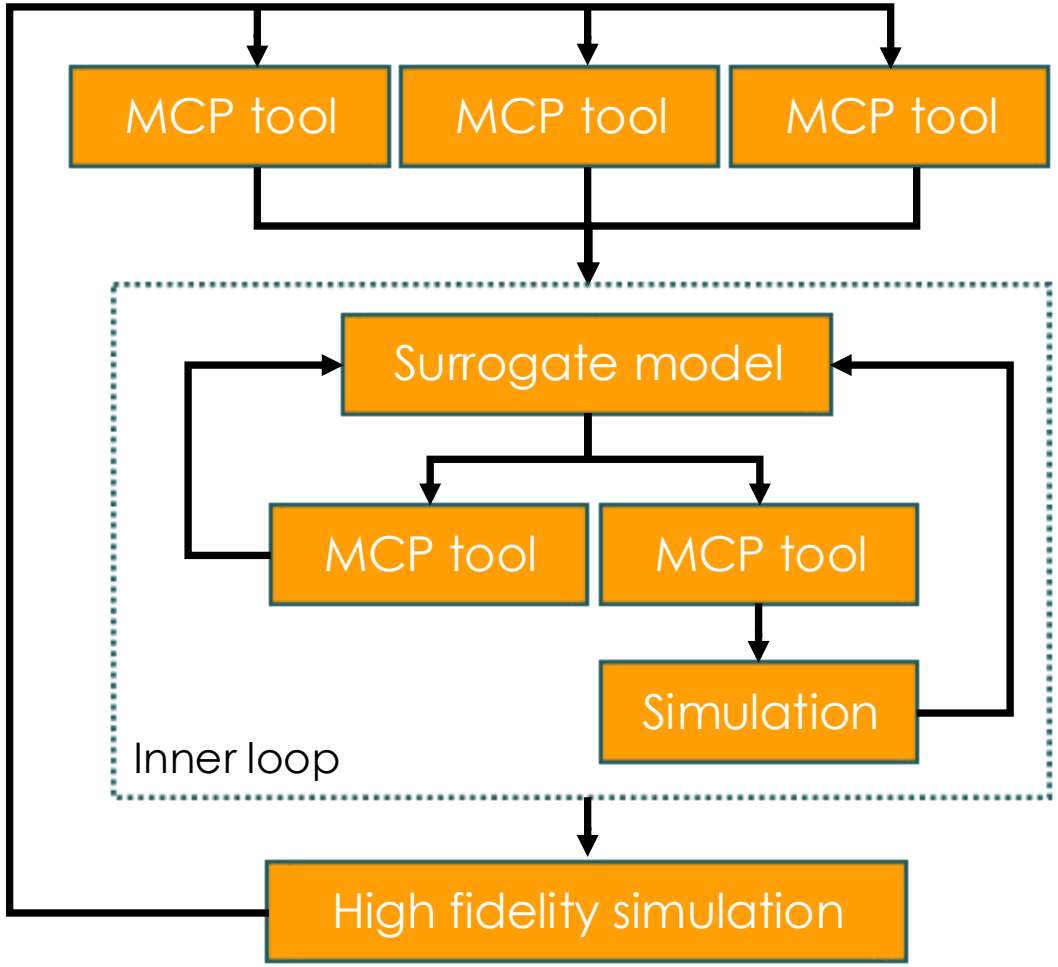
AI applications using reasoning models





HPC schedulers

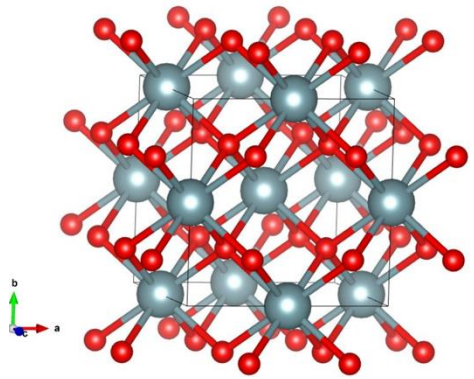
- Two-layer approach
  - Monitoring the tasks and predict the resources needed
  - Task based scheduling with priorities
- Task based scheduling
  - Different priorities per task
  - Separate resources for inner loop and outer loop
    - Sometimes separate resources for the tasks attached to the surrogate model



Finer-grain tasks preferred

Example usage

- Materials discovery
  - New materials to optimize the design of nuclear power plants
  - DFT calculations
    - Heavy high-fidelity simulations
    - Different types of surrogate models



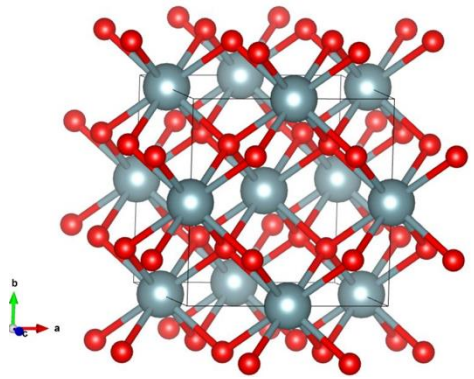
Total energy per atom

Reasoning model example

- Reproduce paper X
- Discover a new material with property X
- Find all stable materials that have property Y

Example usage

- Materials discovery
  - New materials to optimize the design of nuclear power plants
  - DFT calculations
    - Heavy high-fidelity simulations
    - Different types of surrogate models



Total energy per atom

Reasoning model example

Tool to read a paper

Tool to parse a dataset of papers

Custom tools

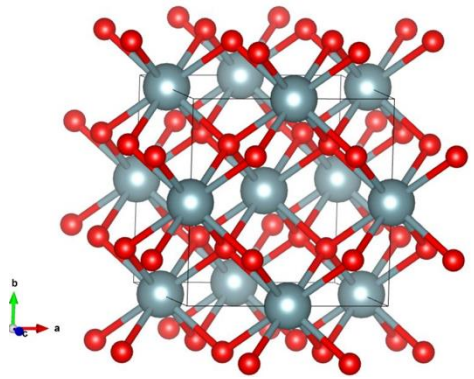
Tool to query into nuclear materials datasets

Tool to find similar materials

MCP tool | MCP tool

Example usage

- Materials discovery
  - New materials to optimize the design of nuclear power plants
  - DFT calculations
    - Heavy high-fidelity simulations
    - Different types of surrogate models



Total energy per atom

Inner loop example

Surrogate models using RF, NN

MCP tool for searching datasets for a given material

MCP tool to find candidate structures for a given composition

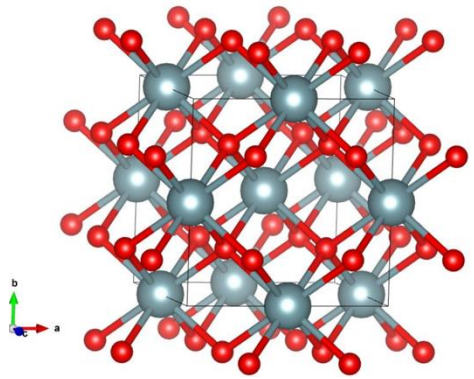
MCP tool to predict energy

DFT computations

MCP tool | MCP tool

Example usage

- Materials discovery
  - New materials to optimize the design of nuclear power plants
  - DFT calculations
    - Heavy high-fidelity simulations
    - Different types of surrogate models



Total energy per atom

Attached tasks

Drift detection for the surrogate models

Drift detection for input data

Continual learning

Visualization

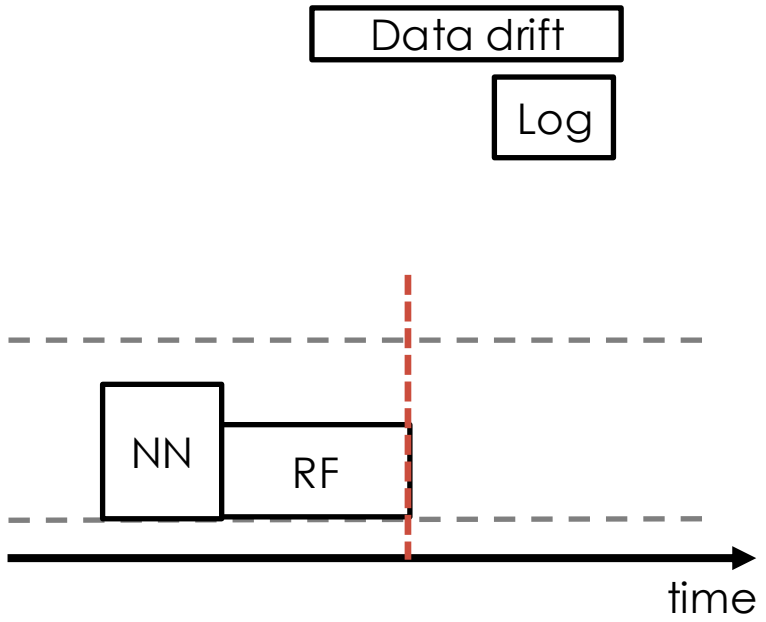
Correctness checks

Logging and update

MCP tool | MCP tool

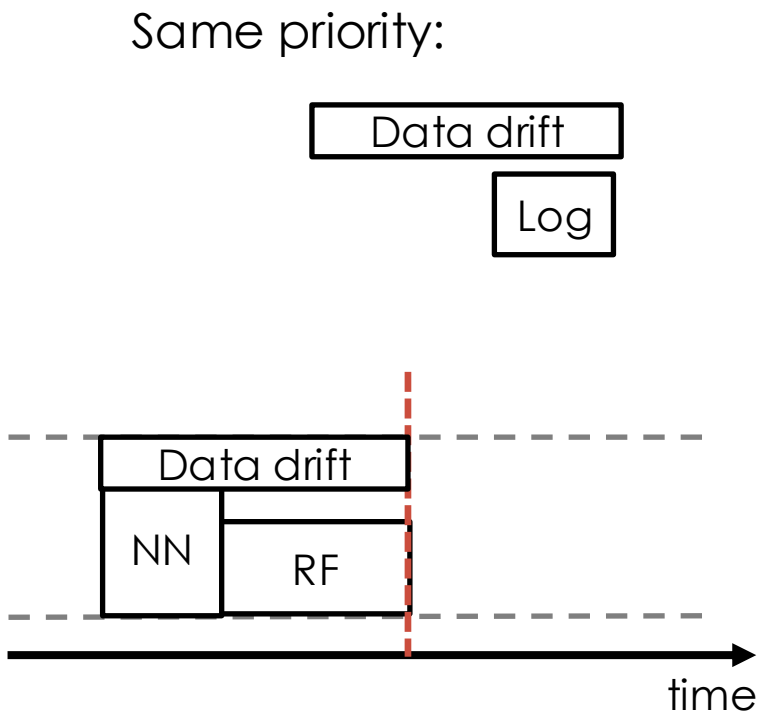
PriorityBF

- Use several priority queues
- Within a queue, jobs are scheduled with an **EASY-BF strategy**
- Between queues, jobs are scheduled **conservatively**
  - Jobs from a queue with a higher index cannot delay jobs with a lower index
- Minimize response times for high-priority jobs



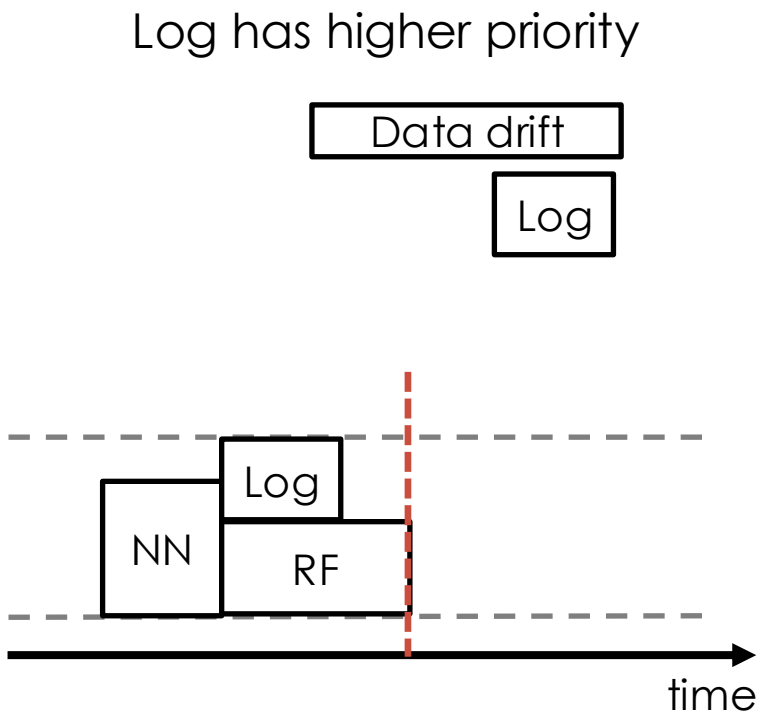
PriorityBF

- Use several priority queues
- Within a queue, jobs are scheduled with an **EASY-BF strategy**
- Between queues, jobs are scheduled **conservatively**
  - Jobs from a queue with a higher index cannot delay jobs with a lower index
- Minimize response times for high-priority jobs



PriorityBF

- Use several priority queues
- Within a queue, jobs are scheduled with an **EASY-BF strategy**
- Between queues, jobs are scheduled **conservatively**
  - Jobs from a queue with a higher index cannot delay jobs with a lower index
- Minimize response times for high-priority jobs



PriorityBF

- Use several priority queues
- Within a queue, jobs are scheduled with an **EASY-BF strategy**
- Between queues, jobs are scheduled **conservatively**
  - Jobs from a queue with a higher index cannot delay jobs with a lower index
- Minimize response times for high-priority jobs

- Propose a stable material for UO2

  Workflow created:

  - Find structures and their enthalpy in available datasets.
  - If no structure exists, generate template structures
  - For each template structure predict the enthalpy using RF and NN
  - Return the entries with lowest enthalpy per atom

PriorityBF

- Tasks
  - High priority
    - Prediction codes (NN, RF, etc)
    - Continual learning
  - Low priority
    - Check correctness, visualize and log

- Goal
  - In a given timeframe, check as many materials as possible
    - Drift detection to keep the model synced
    - Vizualization of the datapoints created
    - Log progress and update parameters

- Propose a stable material for UO2

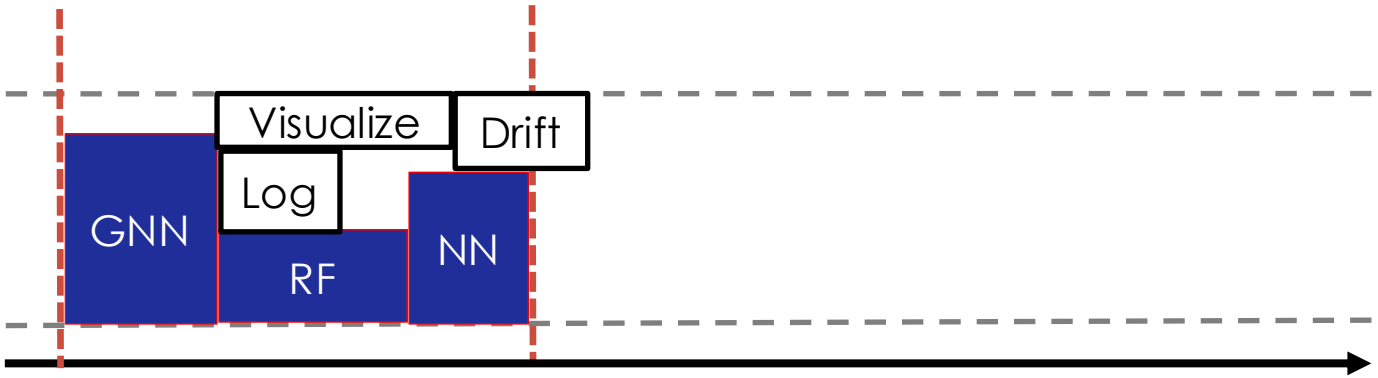
  Workflow created:

  - Find structures and their enthalpy in available datasets.
  - If no structure exists, generate template structures
  - For each template structure predict the enthalpy using RF and NN
  - Return the entries with lowest enthalpy per atom

PriorityBF

All codes predicting the energy of material UO2 with structure S1 have finished

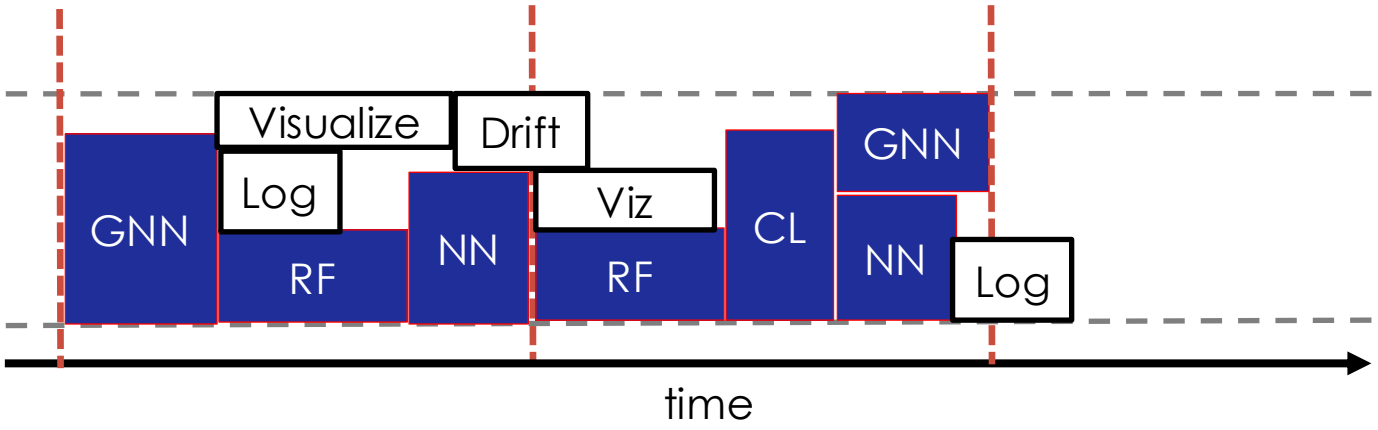
- Release NN, RF, GNN codes for next material (S2)
- Release processing tasks for UO2 with S2
- Remove from the wait queue all tasks related to S1



PriorityBF

All codes predicting the energy of material UO2 with structure S2 have finished

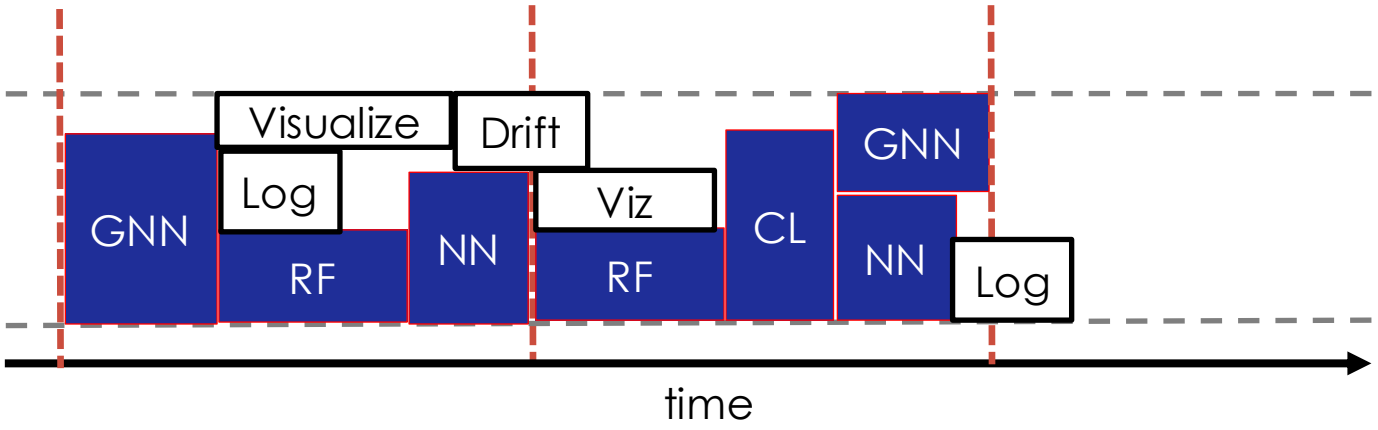
- Drift detected, releases high priority CL code
- Core per process depends on materials



PriorityBF

All codes predicting the energy of material UO2 with structure S2 have finished

- Drift detected, releases high priority CL code
- Core per process depends on materials



- 1124 structures possible for UO2
  - 3 prediction methods
  - Drift detected 6 times for NN
  - Drift detected 3 times for GNN

1.3x – 1.6x more materials investigated in the same time window

Drift checked avg 60% less times
- CL for NN postponed by 12 structures
- CL for GNN postponed by 21 structures

Larger example

Journal of Nuclear Materials 510 (2018) 331–336

Contents lists available at ScienceDirect

Journal of Nuclear Materials

journal homepage: www.elsevier.com/locate/jnucmat

Stability of U_5Si_4 phase in U-Si system: Crystal structure prediction and phonon properties using first-principles calculations

D.A. Lopes*, V. Kocevski, T.L. Wilson, E.E. Moore, T.M. Besmann

Nuclear Engineering Program, Department of Mechanical Engineering, University of South Carolina, Columbia, SC, 29209, USA

ARTICLE INFO

ABSTRACT

- Reproduce the findings of paper X

  Workflow created:

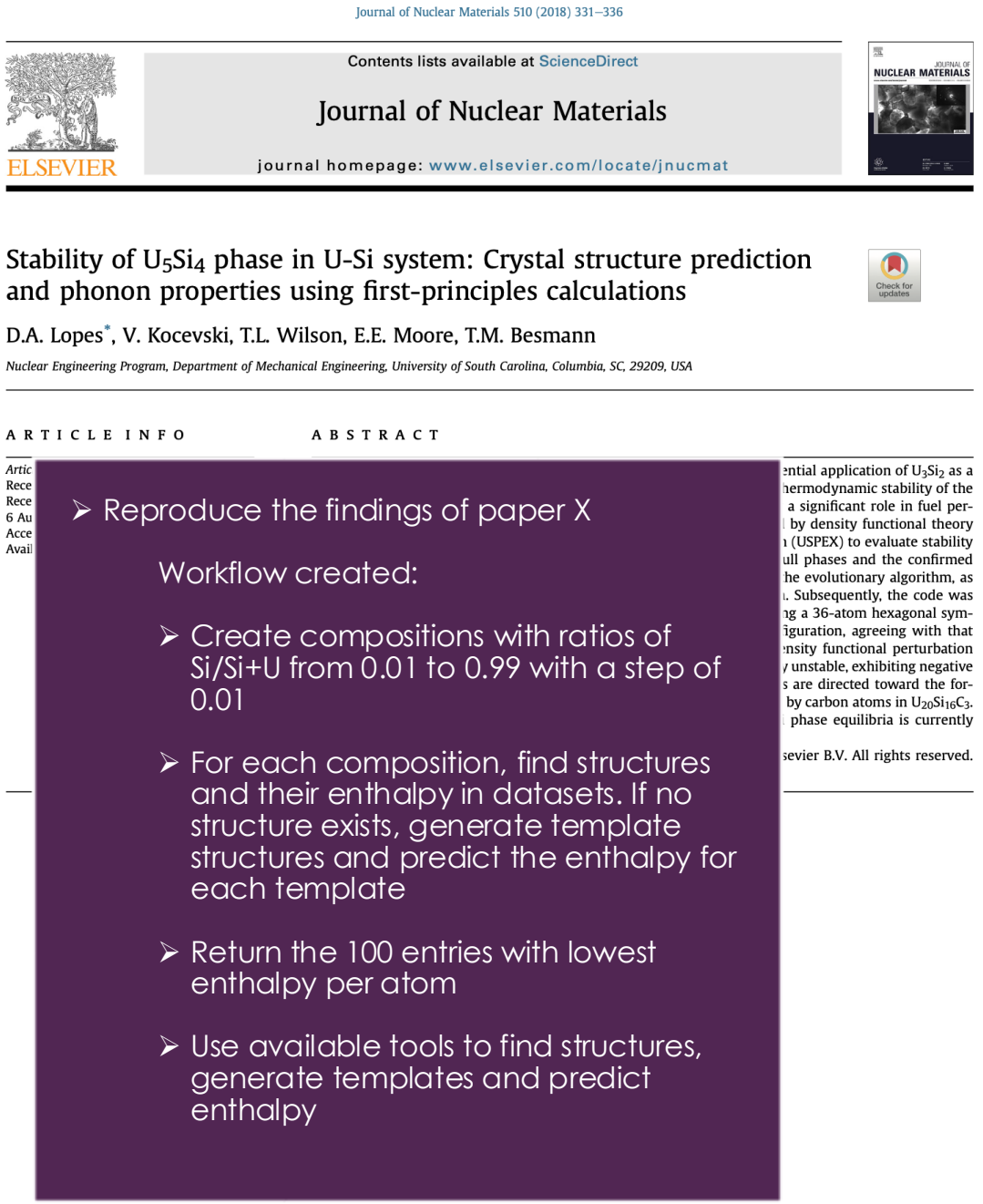
  - Create compositions with ratios of Si/Si+U from 0.01 to 0.99 with a step of 0.01
  - For each composition, find structures and their enthalpy in datasets. If no structure exists, generate template structures and predict the enthalpy for each template
  - Return the 100 entries with lowest enthalpy per atom
  - Use available tools to find structures, generate templates and predict enthalpy

Larger example

- Aprox. 9.5 million materials
- PriorityBF
  - Drift every 10 materials
  - Visualization every 100 materials
  - Validation every 10 materials
  - Logging every 5 materials

Average 1.5x more materials investigated in the same time window

Number of misses for low priority jobs < 100



Journal of Nuclear Materials 510 (2018) 331–336

Contents lists available at ScienceDirect

Journal of Nuclear Materials

journal homepage: www.elsevier.com/locate/jnucmat

Stability of U_5Si_4 phase in U-Si system: Crystal structure prediction and phonon properties using first-principles calculations

D.A. Lopes*, V. Kocevski, T.L. Wilson, E.E. Moore, T.M. Besmann

Nuclear Engineering Program, Department of Mechanical Engineering, University of South Carolina, Columbia, SC, 29209, USA

ARTICLE INFO

ABSTRACT

- Reproduce the findings of paper X

  Workflow created:

  - Create compositions with ratios of Si/Si+U from 0.01 to 0.99 with a step of 0.01
  - For each composition, find structures and their enthalpy in datasets. If no structure exists, generate template structures and predict the enthalpy for each template
  - Return the 100 entries with lowest enthalpy per atom
  - Use available tools to find structures, generate templates and predict enthalpy

Larger example

- Original study
  - 6 month
  - 500+ DFT simulations ran
  - Over 10k node hours
- Reasoning model
  - One week
  - Drift detected <10 times
  - Total < 1k node hours

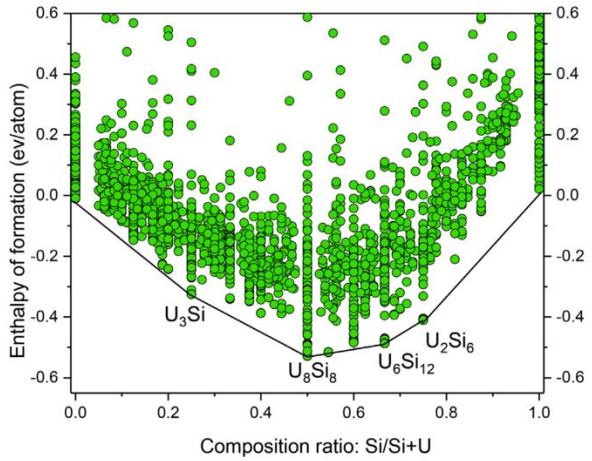
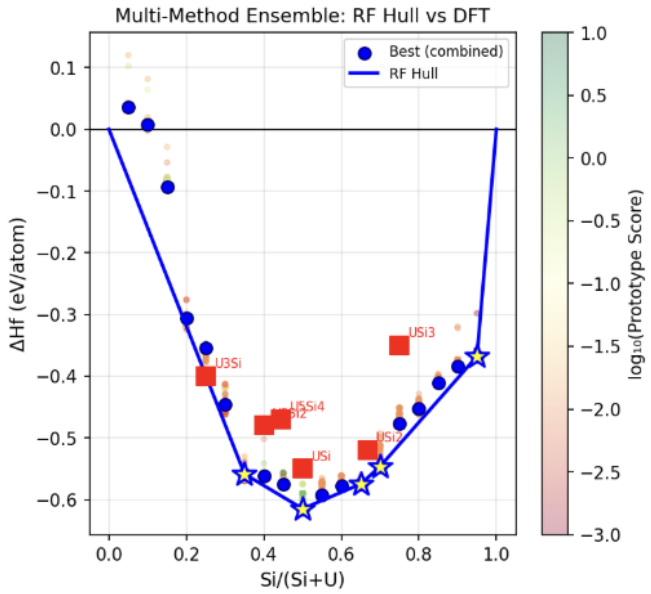


Fig. 2. Convex hull for the U-Si phase space obtained from variable-composition USPEX calculations mode in (enthalpies relative to Si-diamond and α-U, with U_{eff} = 1.5eV), for 2500 structure optimizations.



Journal of Nuclear Materials 510 (2018) 331–336

Contents lists available at ScienceDirect

Journal of Nuclear Materials

journal homepage: www.elsevier.com/locate/jnucmat

Stability of U_5Si_4 phase in U-Si system: Crystal structure prediction and phonon properties using first-principles calculations

D.A. Lopes*, V. Kocevski, T.L. Wilson, E.E. Moore, T.M. Besmann

Nuclear Engineering Program, Department of Mechanical Engineering, University of South Carolina, Columbia, SC, 29209, USA

ARTICLE INFO

ABSTRACT

- Reproduce the findings of paper X

  Workflow created:

  - Create compositions with ratios of Si/Si+U from 0.01 to 0.99 with a step of 0.01
  - For each composition, find structures and their enthalpy in datasets. If no structure exists, generate template structures and predict the enthalpy for each template
  - Return the 100 entries with lowest enthalpy per atom
  - Use available tools to find structures, generate templates and predict enthalpy

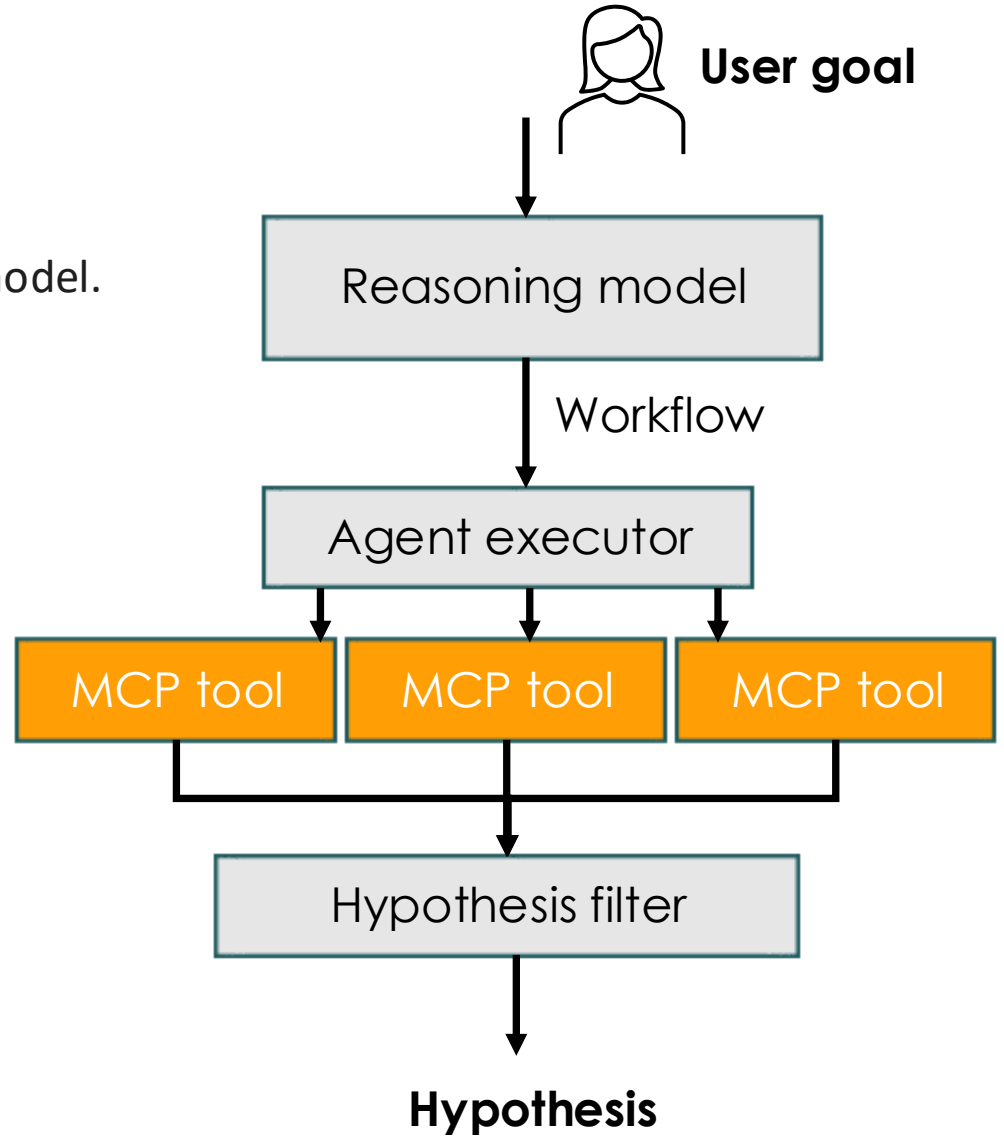
OAK RIDGE National Laboratory

Token usage

Input Tokens	Your prompt + any context provided.
Reasoning Tokens	The hidden "thinking" process generated by the model.
Visible Tokens	The final answer shown to the user.

All gray boxes consume tokens.

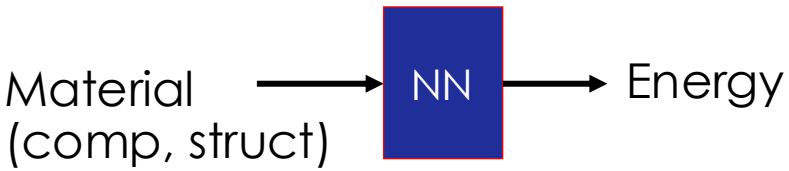
- **Reasoning Model**: Consumes tokens to propose Hypothesis A.
- **Surrogate Model**: Consumes compute to score Hypothesis A.
- **Reasoning Model**: Consumes Input Tokens (to read the score) + Reasoning Tokens (to process the failure/success) + Output Tokens (to propose Hypothesis B)



Task granularity vs token consumption

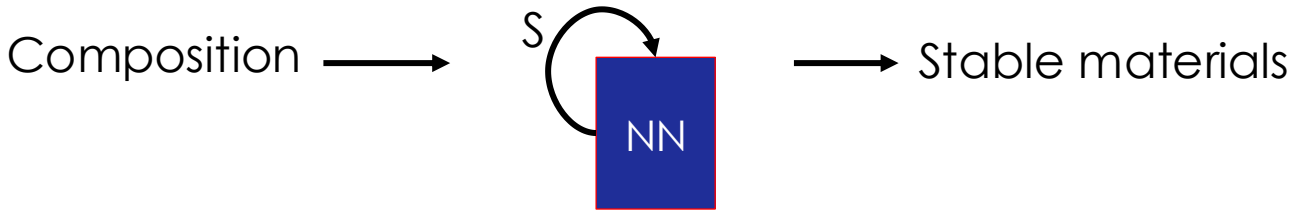
- Fine grain tasks require more login in between calls
  - Higher token consumption

Tool used for finding stable materials



Propose 1 hypothesis -> Check

Tool used for finding all stable materials for a composition



Propose N hypotheses -> Check all via surrogate -> Feed N results back at once.

Token decrease by 98% without an increase in compute

Task granularity vs token consumption

- Fine grain tasks require more login in between calls
  - Higher token consumption

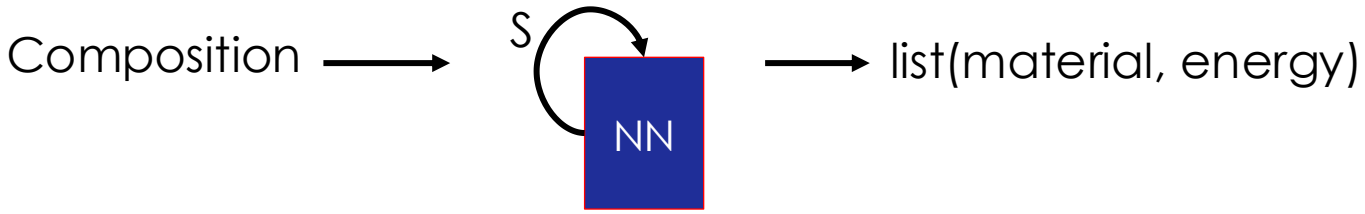
- Compute the energy for material (C, S)

  Workflow created:

  - Run tool to predict energy for C
  - Find entry for (C, S) and return energy

Compute increases by 98%

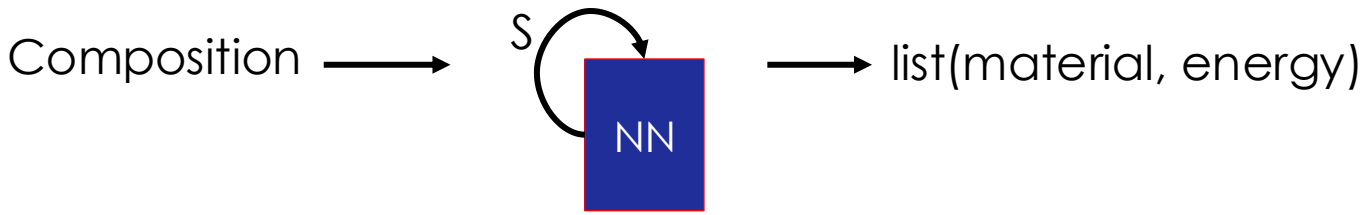
Tool used for finding all stable materials for a composition



Propose N hypotheses -> Check all via surrogate -> Feed N results back at once.

Task granularity vs token consumption

- For each chain of tasks
  - Compute utility for input X for each tool (from 0 to 1)
    - E.g. If a logic step requires 10 predictions, the tool utility is 10/S
  - Compute token magnitude for using each possible tool
    - E.g. If the chain requires calling the tool 10 times, the tool token consumption is 10x
    - E.g. if the tool requires preparing data, tokens could be higher
- Scheduling
  - Choose the tools to run to minimize compute
  - In a given token budget



Task granularity vs token consumption

- Results very dependent on the tasks available and the hypothesis chain
  - We use simulations to choose which tools to give the reasoning model

> Given a set of required outputs, a maximum total token budget, and a fixed pool of computational resources, determine the combination of tasks that produces all required outputs while minimizing the total FLOPs consumed.

Multiple-Choice Knapsack Problem

- We are forced to produce every required output.
- For each output, we must pick one option from its dedicated list

Required Outputs: $B = \{b_1, b_2, \ldots, b_N\}$.

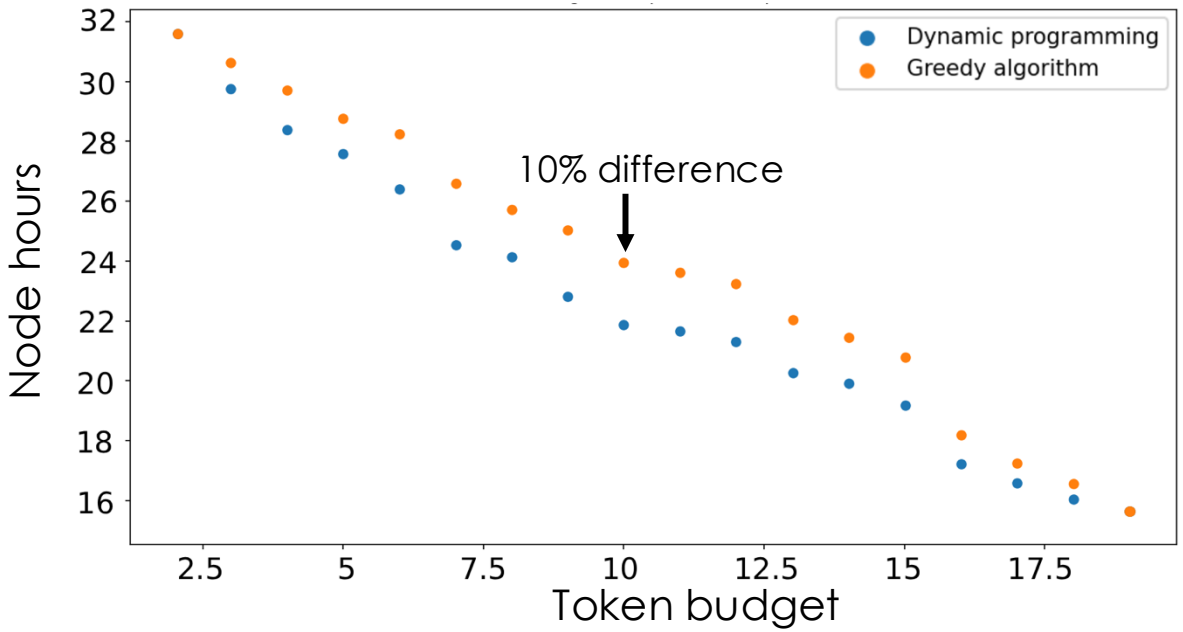
For every b_i, G_i contains all candidate tasks that can produce it.
$G_i = \{t_{i1}, t_{i2}, \ldots, t_{ik}\}$

$U(t_{ij})$: The utility of task j for output I
$T(t_{ij})$: The Tokens consumed by this specific task.

Task granularity vs token consumption

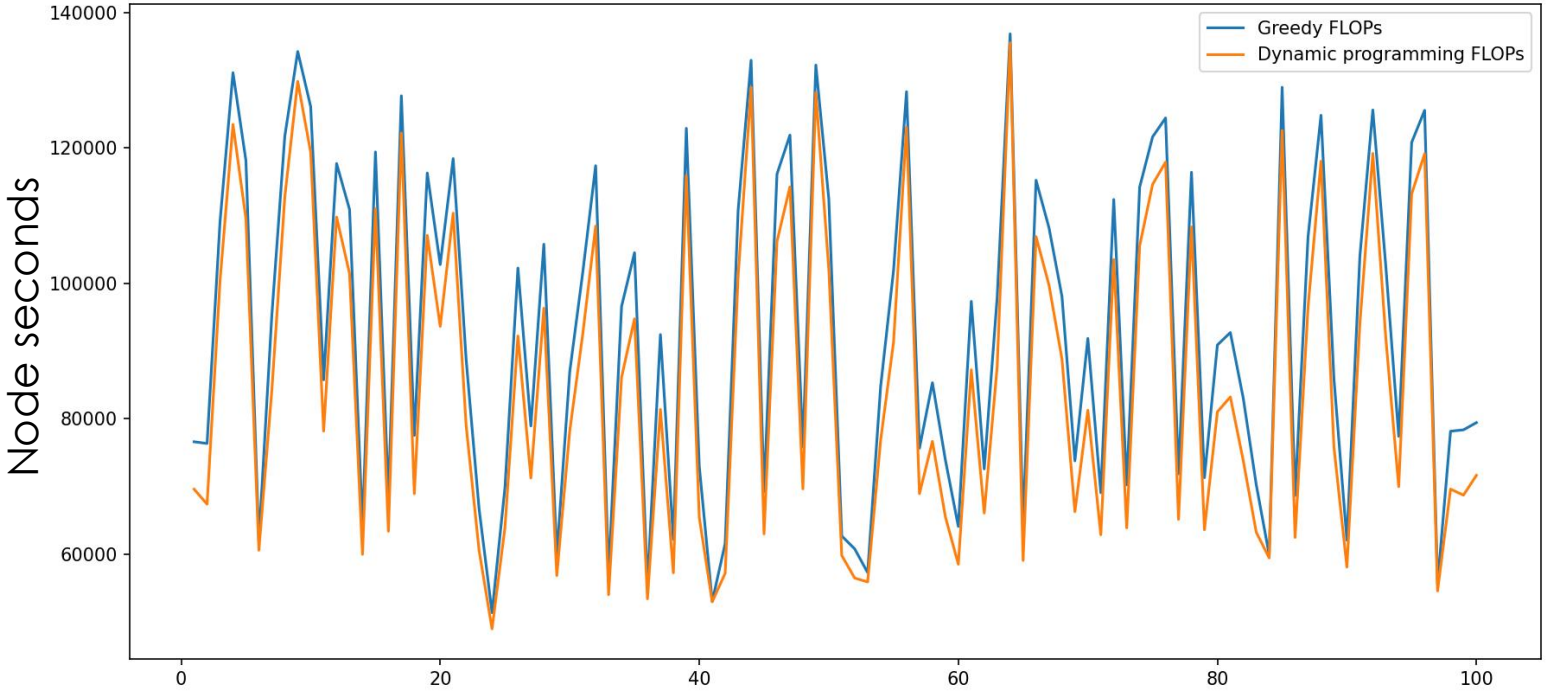
- Greedy algorithm
  - Loop over outputs
    - Start with tasks with max utility
    - When budget is exceeded go back and choose a task with the max utility from what is left (out of all the previous outputs)
  - At the end: list of tasks to achieve all outputs
    - Within the given token budget
    - Maximizing the utility (minimizing FLOPS)

Same results for the fine/coarse grain problem for materials discovery

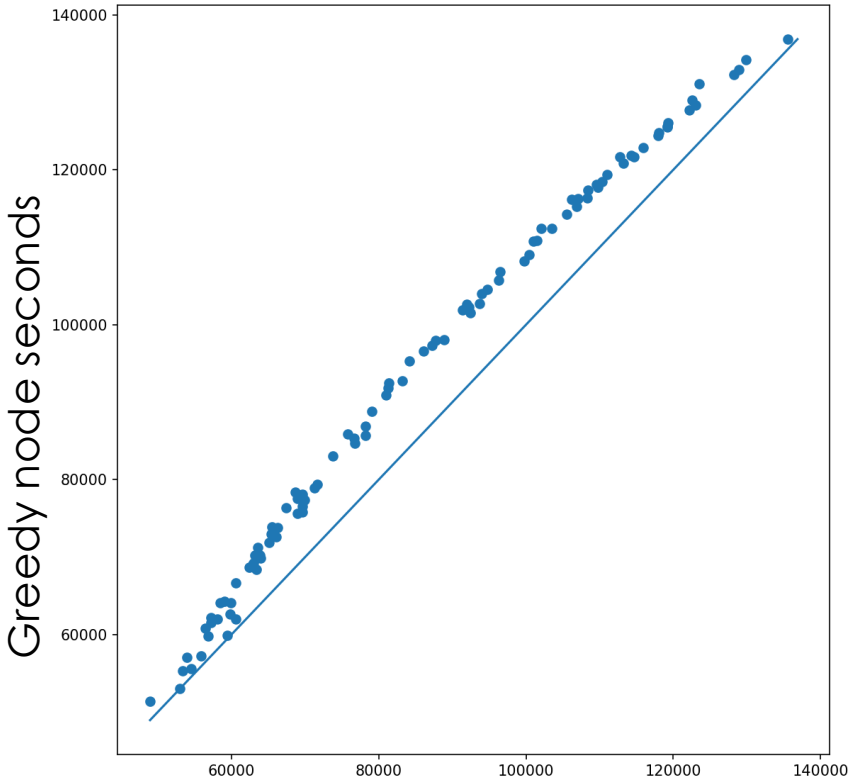


Task granularity vs token consumption

Greedy vs Dynamic programming



Random scenarios of 100 outputs with an average 10 granularity tasks each



Conclusion

- ~~Do not sure LLMs to run workflows~~
- PriorityBF
  - Gives good results for scheduling the computational dynamic tasks created by ad-hoc studies
  - It's better if frequency can be defined
    - Drift detection is not high-priority but the task cannot be starved
  - Finer grain tasks are better
  - More test cases are needed
- Reasoning models
  - Behave better with less freedom
  - Coarse tasks defined specifically for a study are better
  - More testing is needed