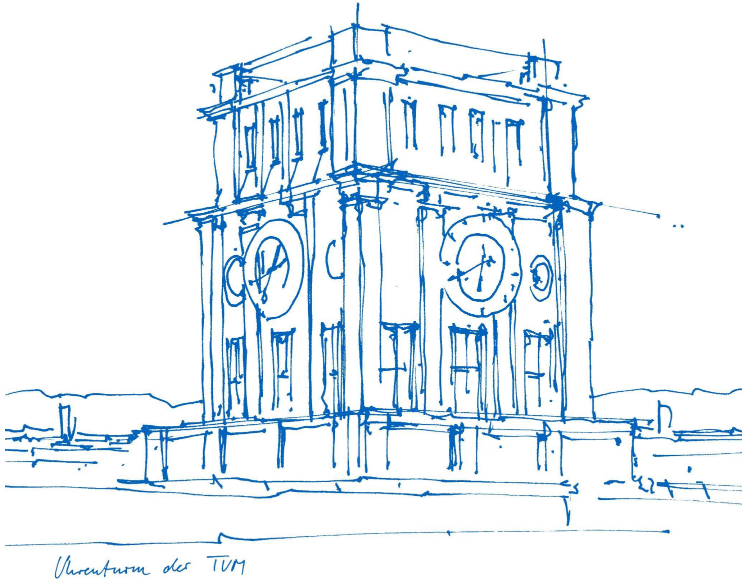
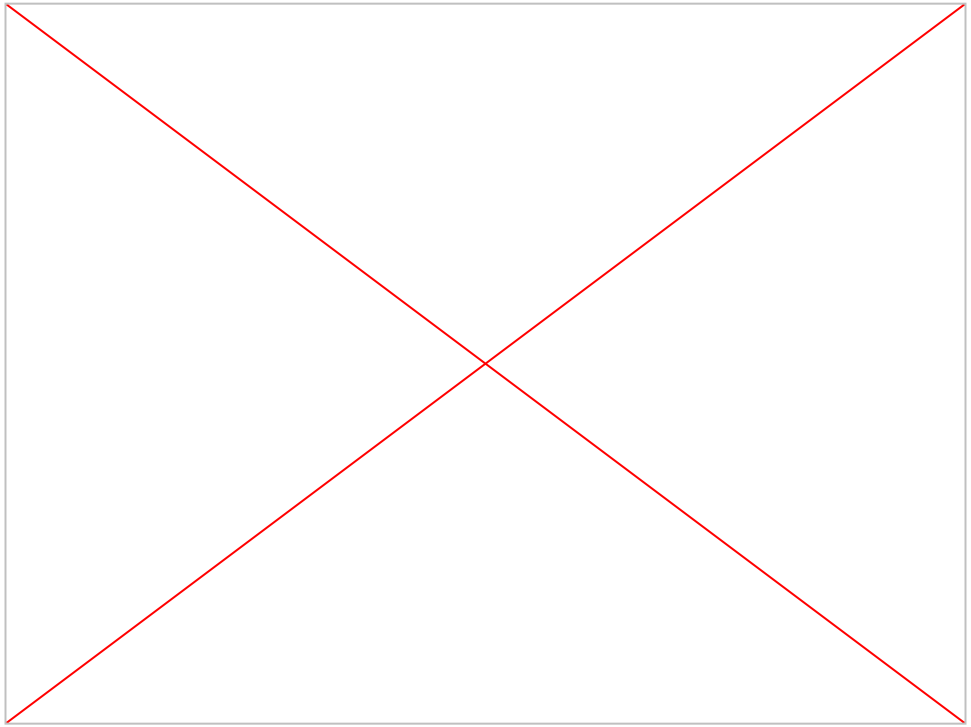
# Algorithm Selection Strategies for Short-Range Particle Simulations

Samuel J. Newcome
Chair of Scientific Computing in Computer Science
Technical University of Munich

Leogang, Austria
25th February 2025

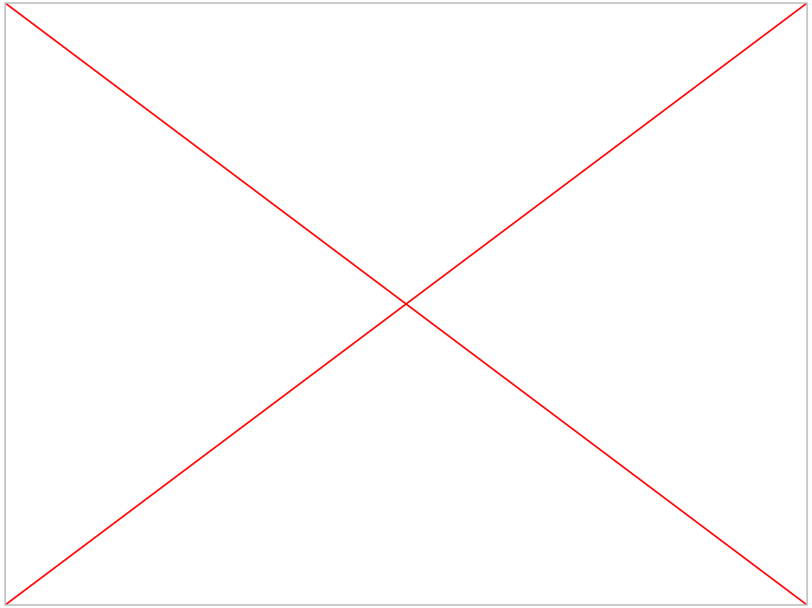


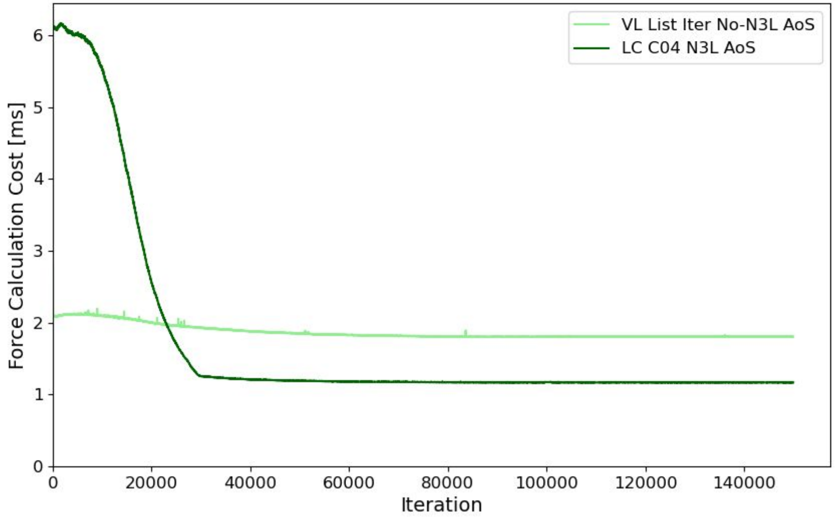
## A Heated Sphere



Technical University of Munich

# A Heated Sphere

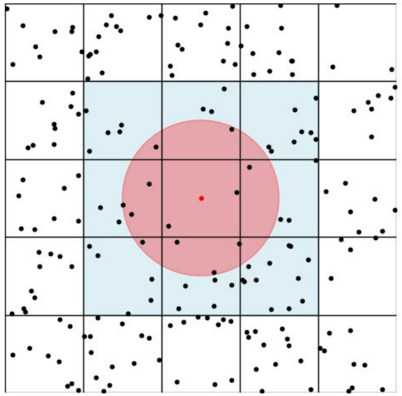




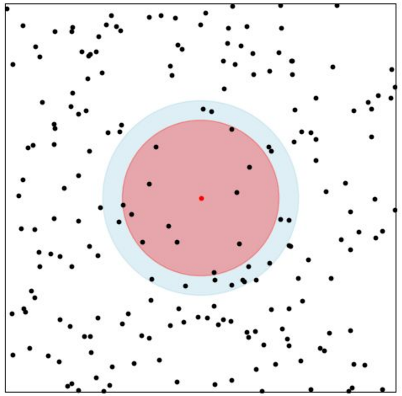
## Short-Range Particle Simulation Methods

To simulate this efficiently, we need a:

- Neighbour Identification Algorithm/Particle Container. E.g.



Linked Cells



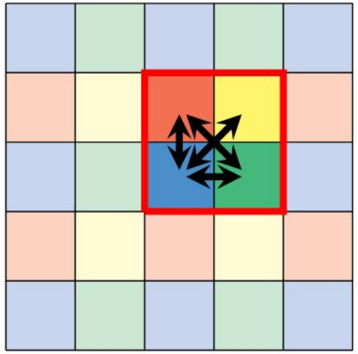
Verlet Lists

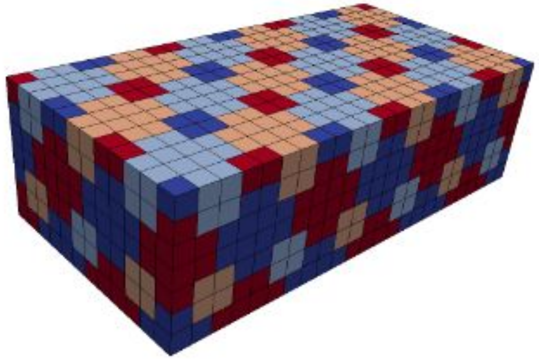
Images from Newcome et al., 2023

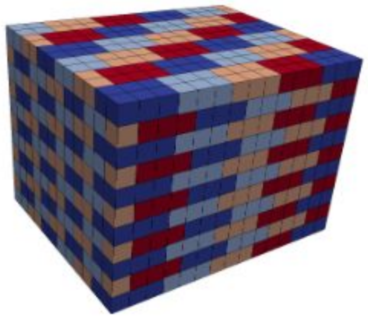
# Short-Range Particle Simulation Methods

To simulate this efficiently, we need a:

- Neighbour Identification Algorithm/Particle Container.
- Shared Memory Traversal e.g. cell colouring schemes


C08


C04


C04_HCP

C08 Image from Newcome et al., 2023
C04 and C04_HCP Images are from Tchipev, 2020

## Short-Range Particle Simulation Methods

To simulate this efficiently, we need a:

- Neighbour Identification Algorithm/Particle Container.
- Shared Memory Traversal.
- Data Layout e.g. Array-of-Structures or Structure-of-Array

## Short-Range Particle Simulation Methods

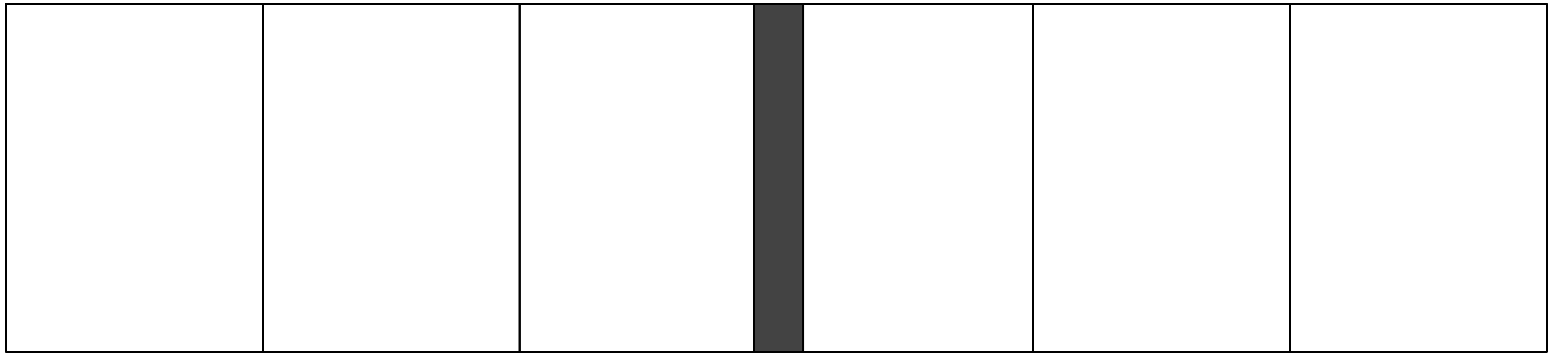
To simulate this efficiently, we need a:

- Neighbour Identification Algorithm/Particle Container.
- Shared Memory Traversal.
- Data Layout e.g. Array-of-Structures or Structure-of-Array

We also have various parameters to tune:

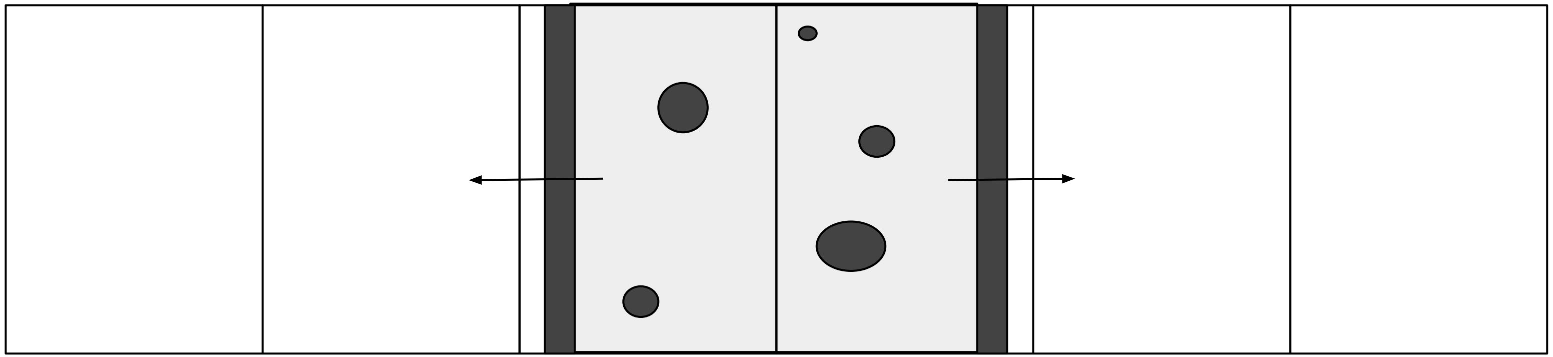
- Size of Cells
- Verlet Skin size

## An Exploding Liquid



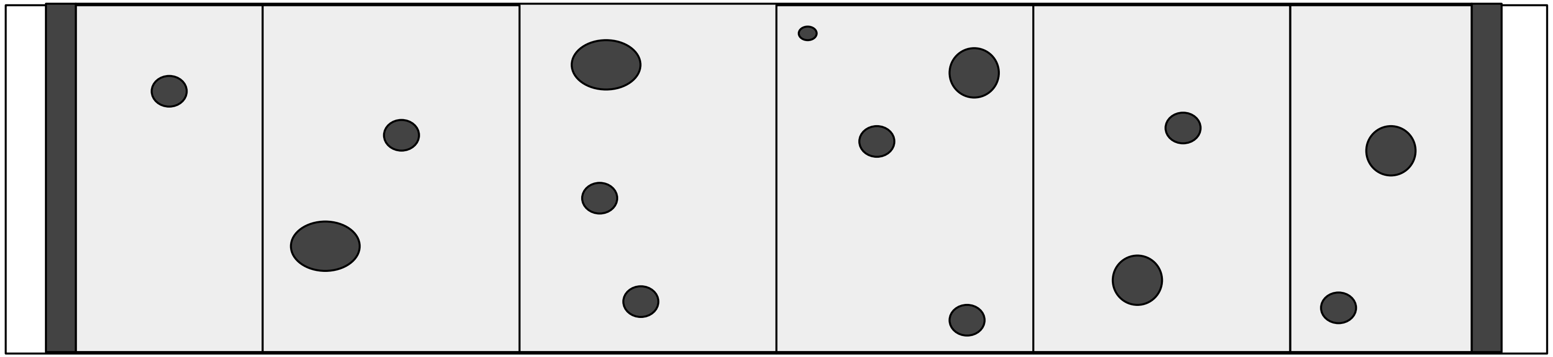
A thin slice of molecules is placed at the centre of a long domain split into 6 MPI ranks.

# An Exploding Liquid



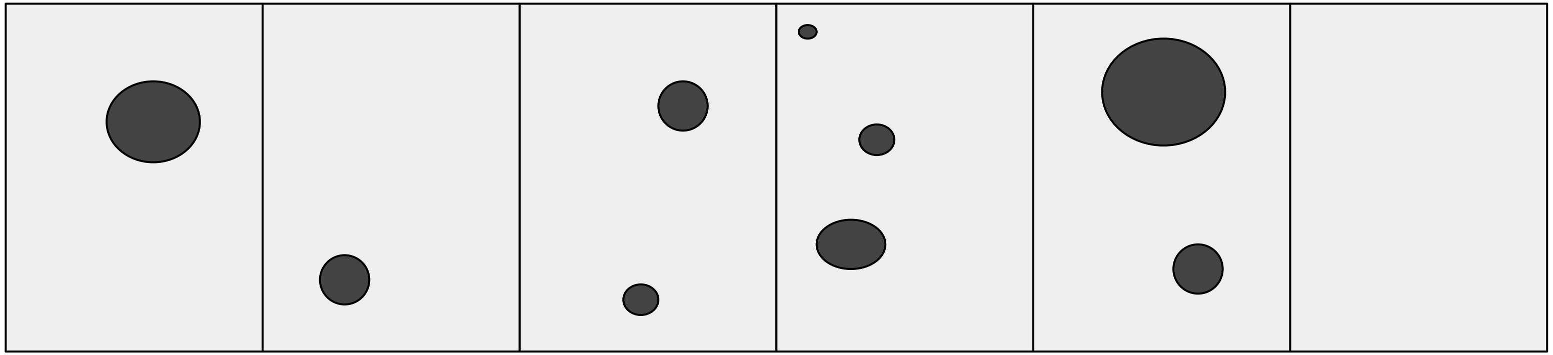
It explodes outwards, leaving behind small clusters of molecules.

# An Exploding Liquid



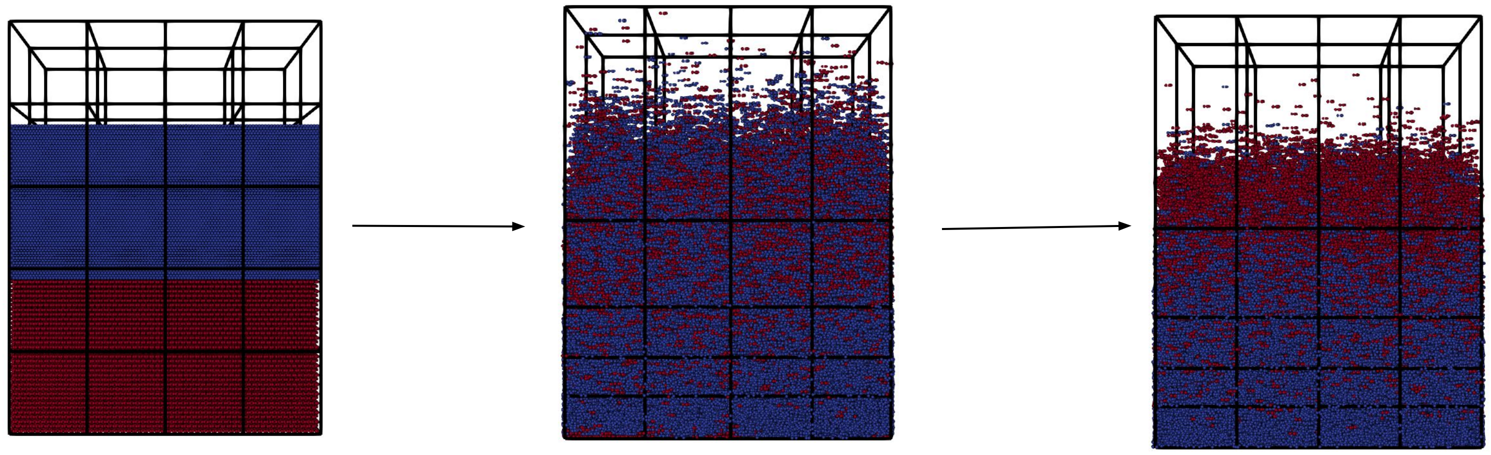
There are different computational profiles in each region, leading to different best algorithms that change over the course of the simulation

## An Exploding Liquid



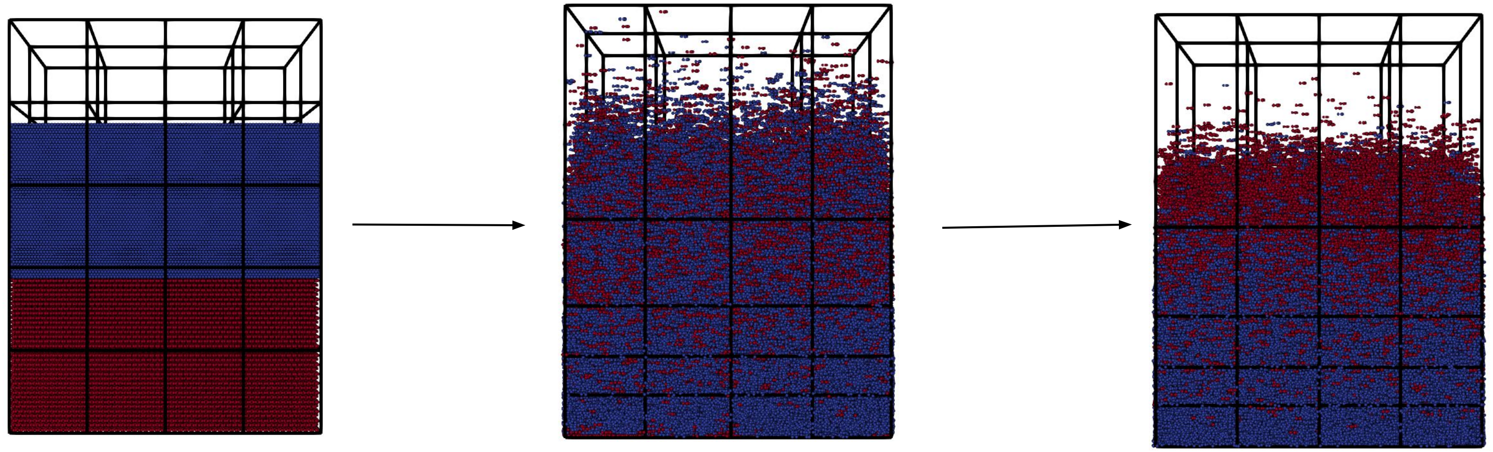
It explodes outwards, leaving behind small clusters of molecules.

## Rayleigh-Taylor Instability



- Simulation starts with “blue” molecules of higher mass and density and smaller size above the “red” molecules.
- 40 MPI ranks are used with MPI load balancing.

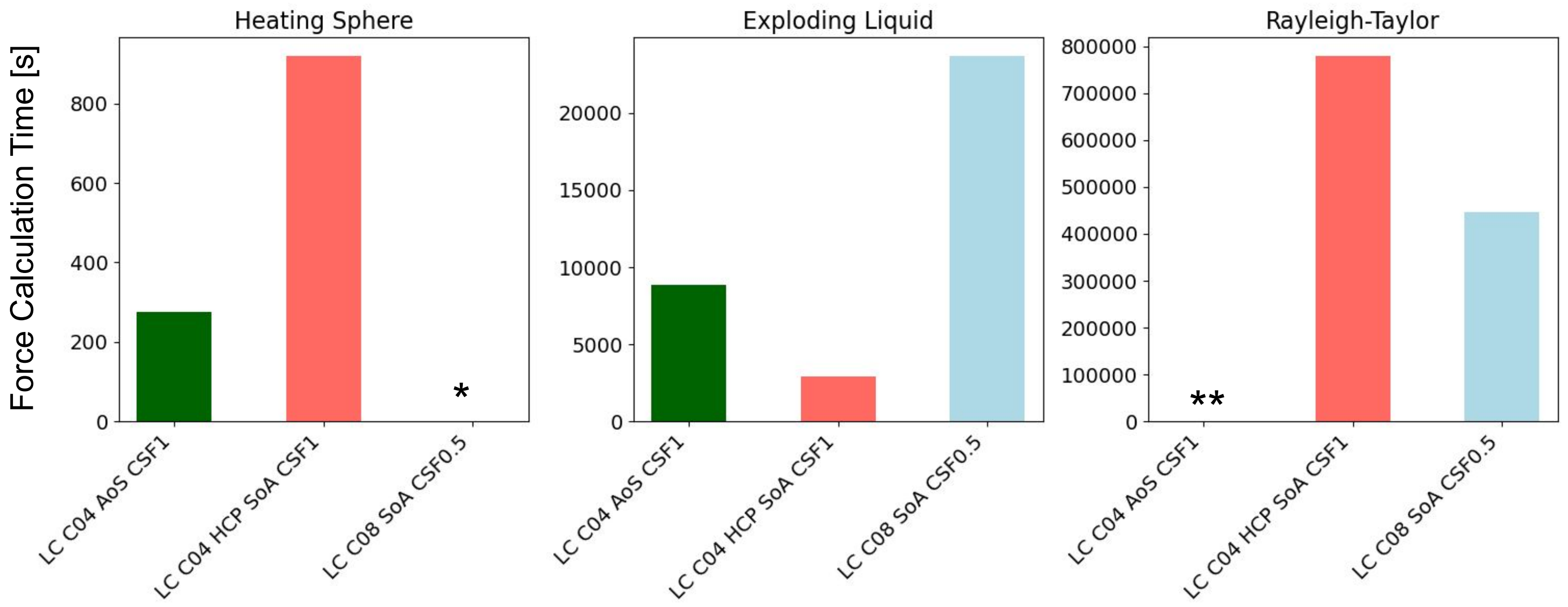
## Rayleigh-Taylor Instability



- The optimal algorithmic configuration is different depending on the mixture of red/blue particles and empty space, and therefore also changes over time and in different regions.

Technical University of Munich

# No “Silver Bullet”



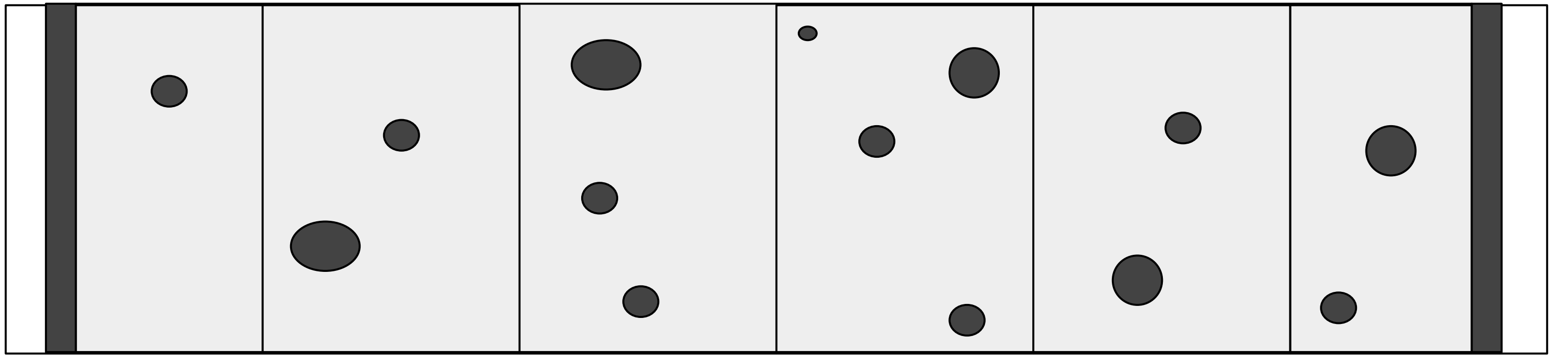
=> There is no best algorithmic configuration

* This configuration timed out. Expected 1-2 order of magnitude worse than best.
** This experiment is still running, but currently expected ~1 order of magnitude worse than best.

# AutoPas: A Rank-Level Algorithm Selection Library

- General **black-box** short-range particle simulation library.
- Users can build their simulator by providing a particle class and an interaction functor class. They don’t need to choose an optimal algorithm.
- 100+ configurations & growing
  - Neighbour Identification Algorithms
  - Shared Memory Parallel Traversals
  - Data Layouts
  - Tunable Parameters e.g. cell-size factor

# AutoPas & Distributed Memory



- AutoPas is rank-level
- Each rank get its own AutoPas container and can make its own choices.

## Algorithm Selection

- AutoPas periodically makes algorithm selection choices every number of timesteps. We call the selection process a *tuning phase*.
- By default, in each tuning phase, each algorithm is trialled for a few iterations.
- The best is used until the next tuning phase.
- Best = Fastest Time or Least Energy Consumed
- No accuracy difference between algorithms => timesteps spent trialling can still advance the simulation.

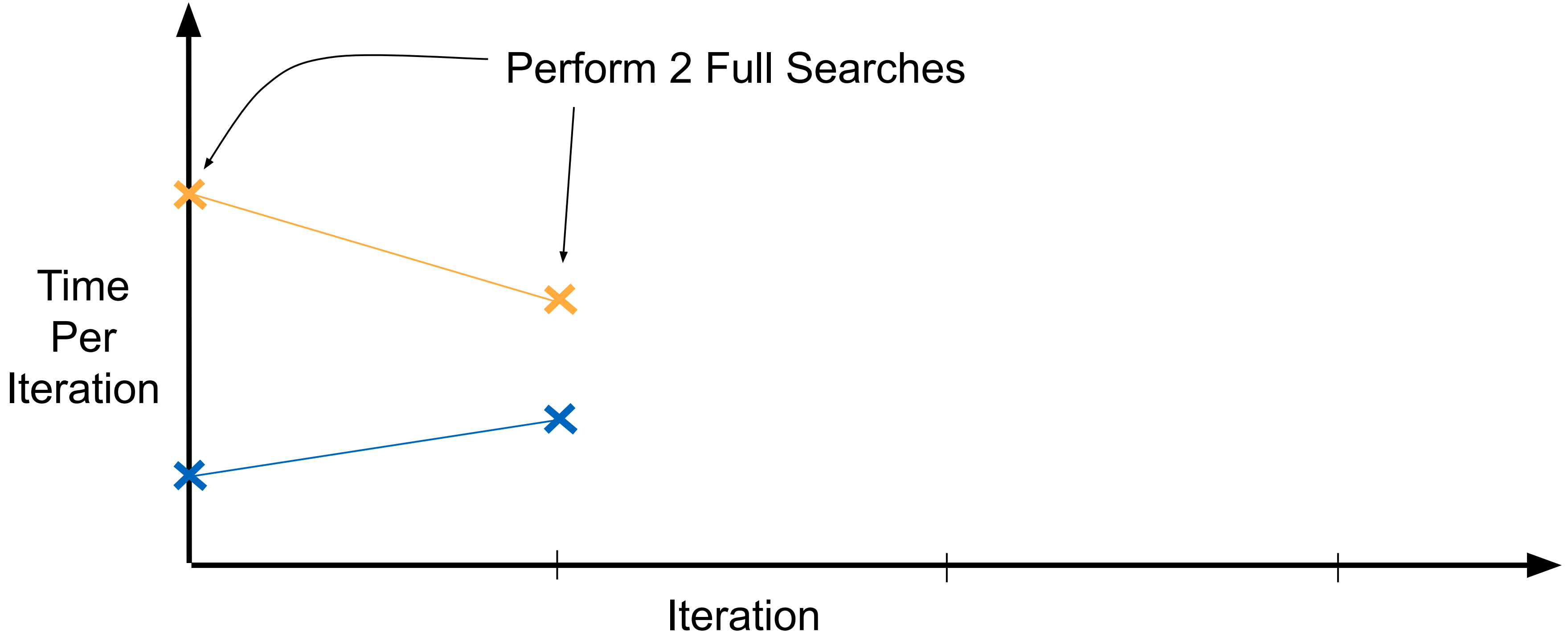
Algorithm Selection

As previously seen, the worst algorithms can be orders of magnitude worse.

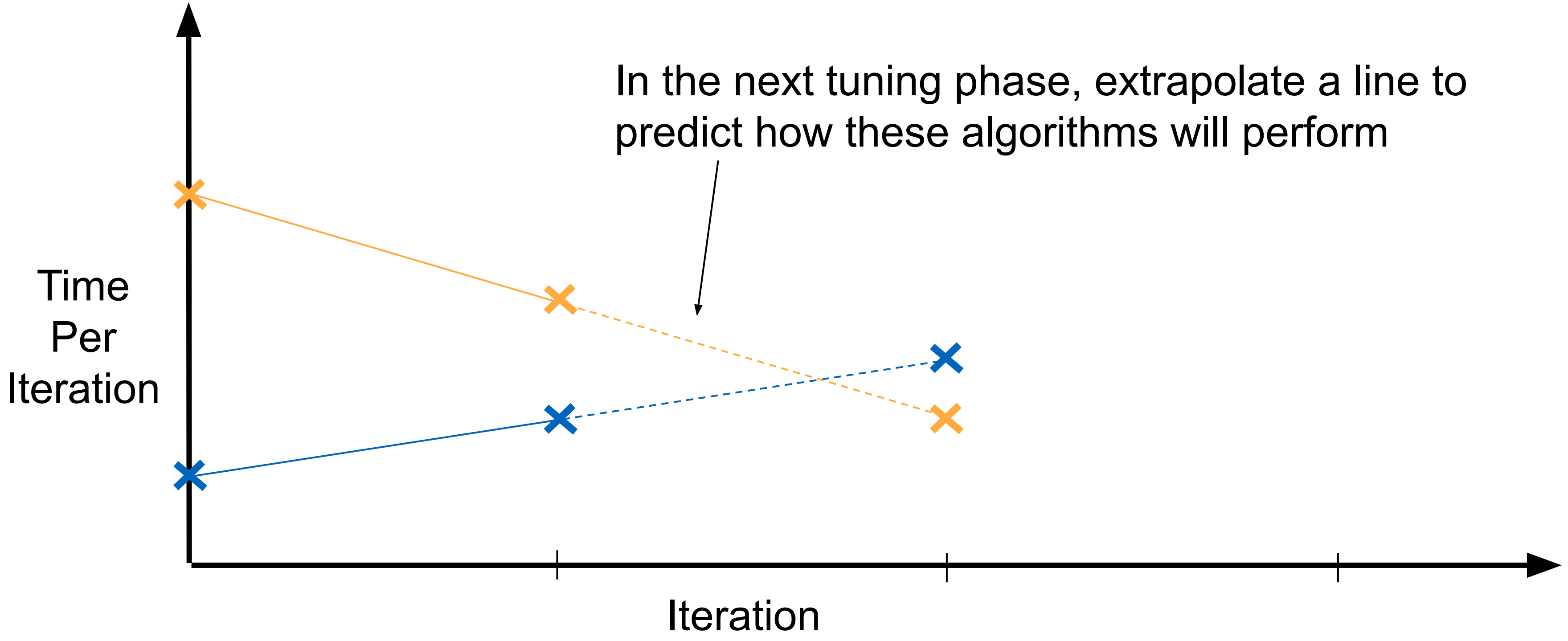
=> We need methods to avoid trialling bad algorithms.

## Algorithm Selection Strategies

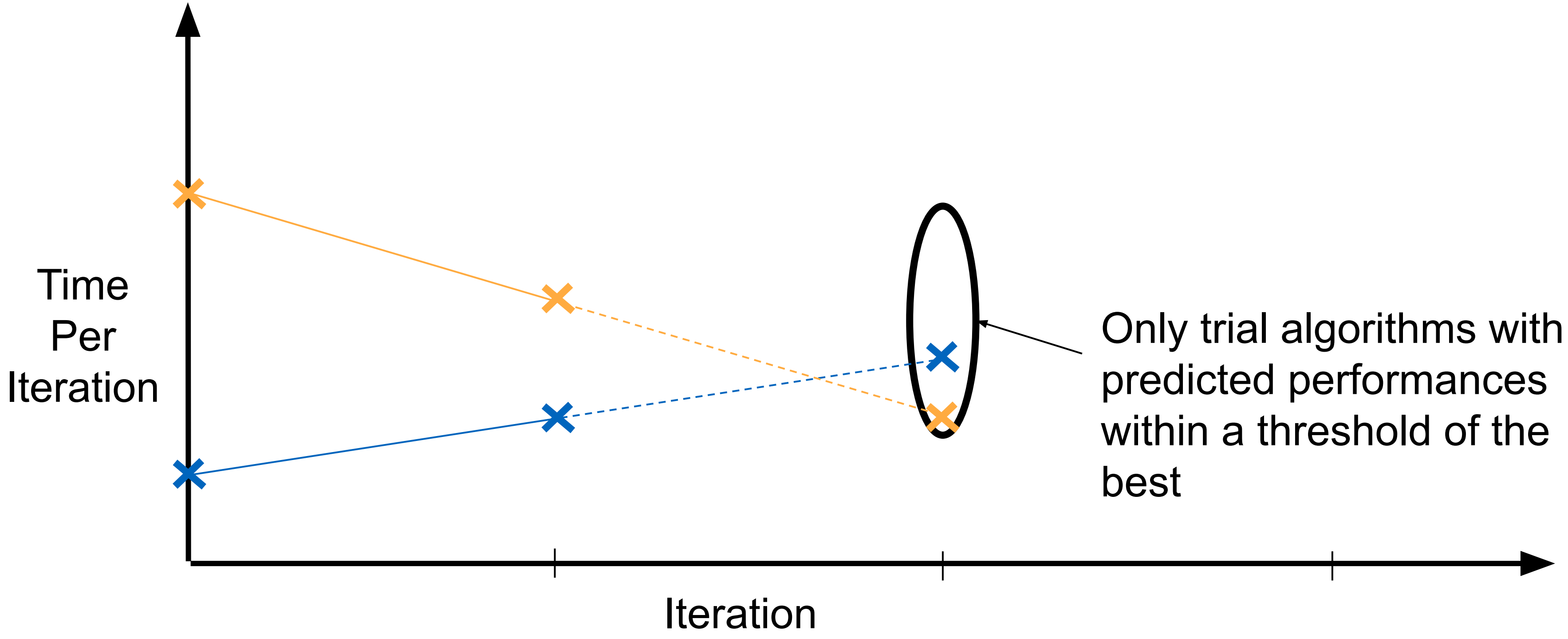
## Predictive Tuning



## Predictive Tuning

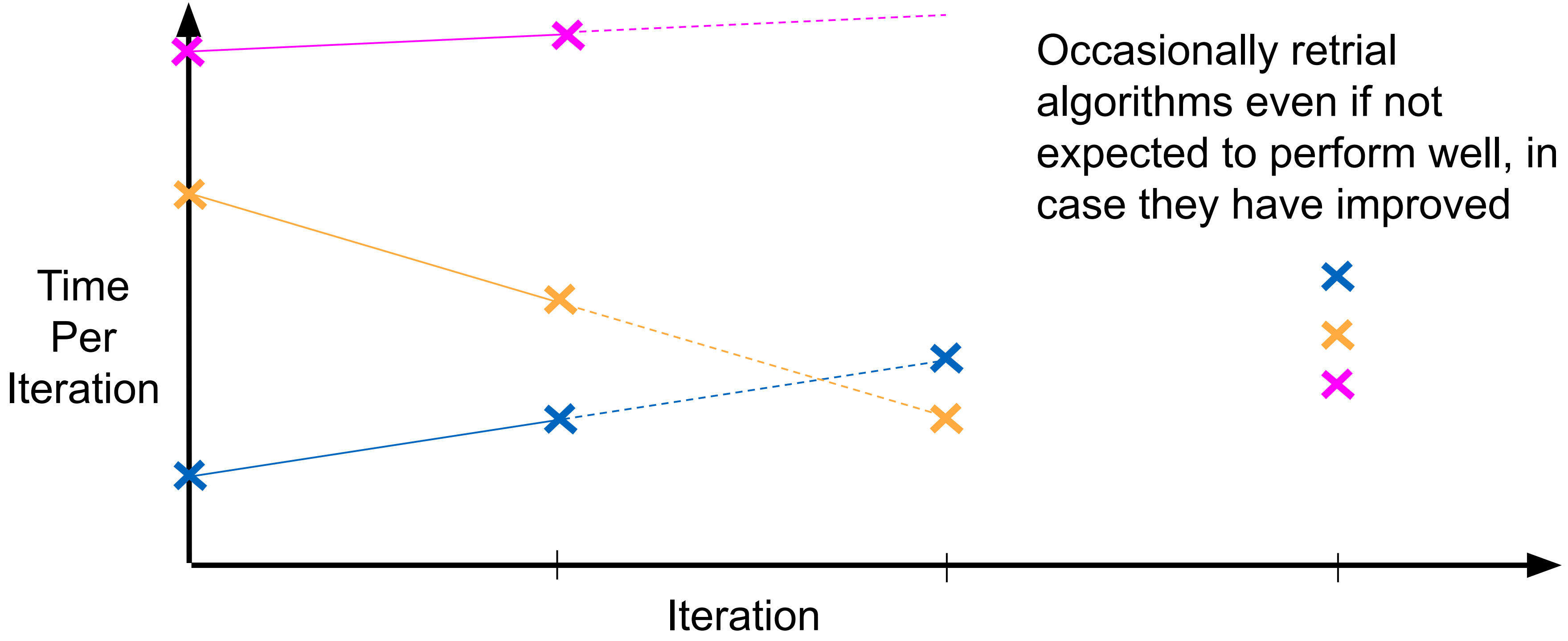


# Predictive Tuning



Only trial algorithms with predicted performances within a threshold of the best

# Predictive Tuning



Occasionally retrial algorithms even if not expected to perform well, in case they have improved

## Predictive Tuning

Pros:

- Easy to use (requires no user input or training).
- => Generalises easily to any arbitrary user simulator.

Cons:

- Predictions can be very unsuitable.
- Requires some naive full searches.

## Expert-Knowledge Fuzzy Logic Tuning

- Take “information” from simulation: E.g.
  - Mean number of particles per cell
  - Median
  - Standard Deviation
- An expert develops (fuzzy logic) rules to describe how suitable a method is depending on this information.
- If a method passes a suitability threshold, it will be trialled.

Technical University of Munich

## Expert-Knowledge Fuzzy Logic Tuning

Pros:

- Highly performant with suitable rules.
- Use of fuzzy logic helps realise “fuzzy” understandings of relationships between statistics and algorithm performance.

Cons:

- Relative algorithm performance varies between interaction models and hardware => A universal set of rules is not feasible
- Relationship between statistics and best algorithm is highly complex => Requires a lot of human-effort even with same model and hardware.

=> Develop rules that are targeted towards the type of simulations being run.

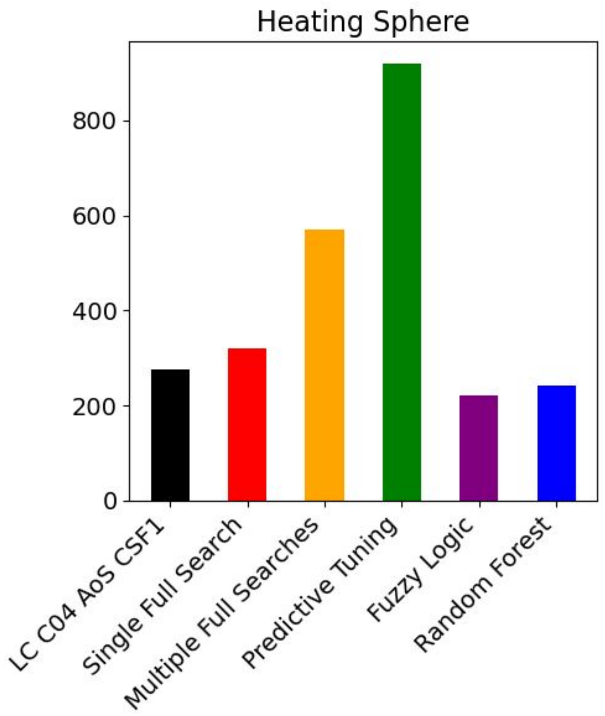
## Random Forest Tuning

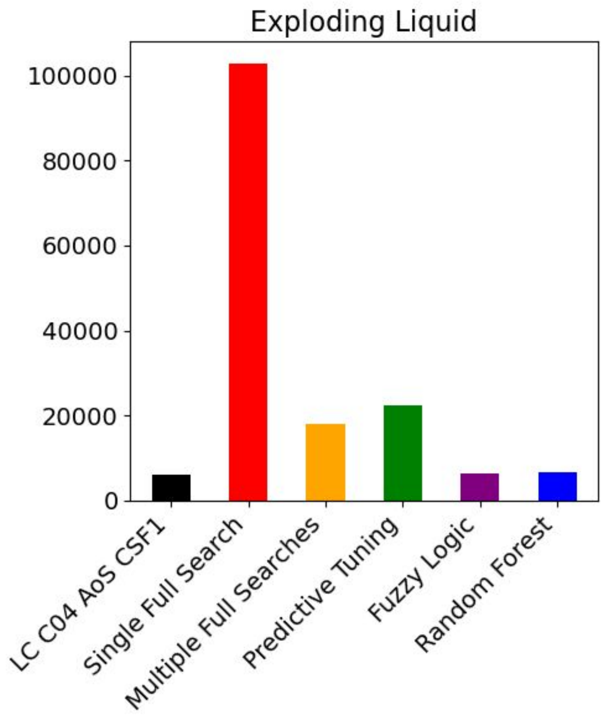
- Train a Random Forest model that predicts optimal algorithm from statistics.
- Random Forest to deal with overfitting of Decision Trees.
- Provides single supposedly optimal configuration.
- Implemented through C/Python API to aid in extensibility.
- Trained on “fake” simulations generated with (random) statistical distributions that are easy to obtain.

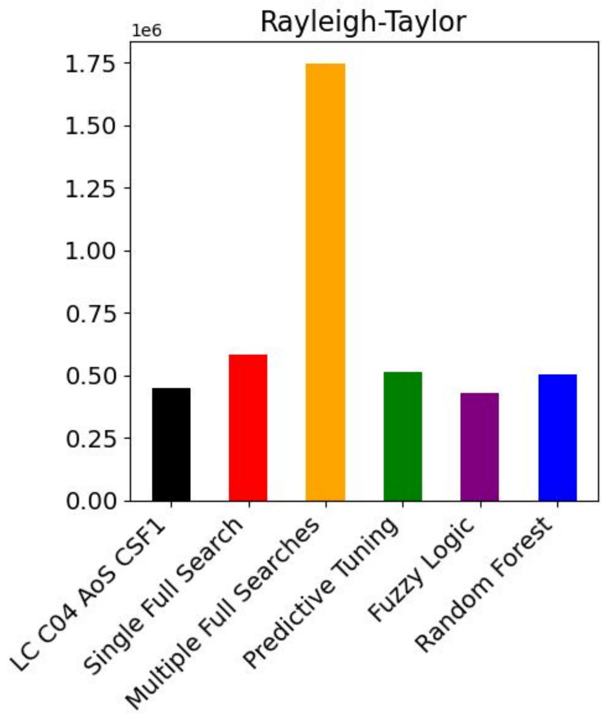
Technical University of Munich

# Results

# Results

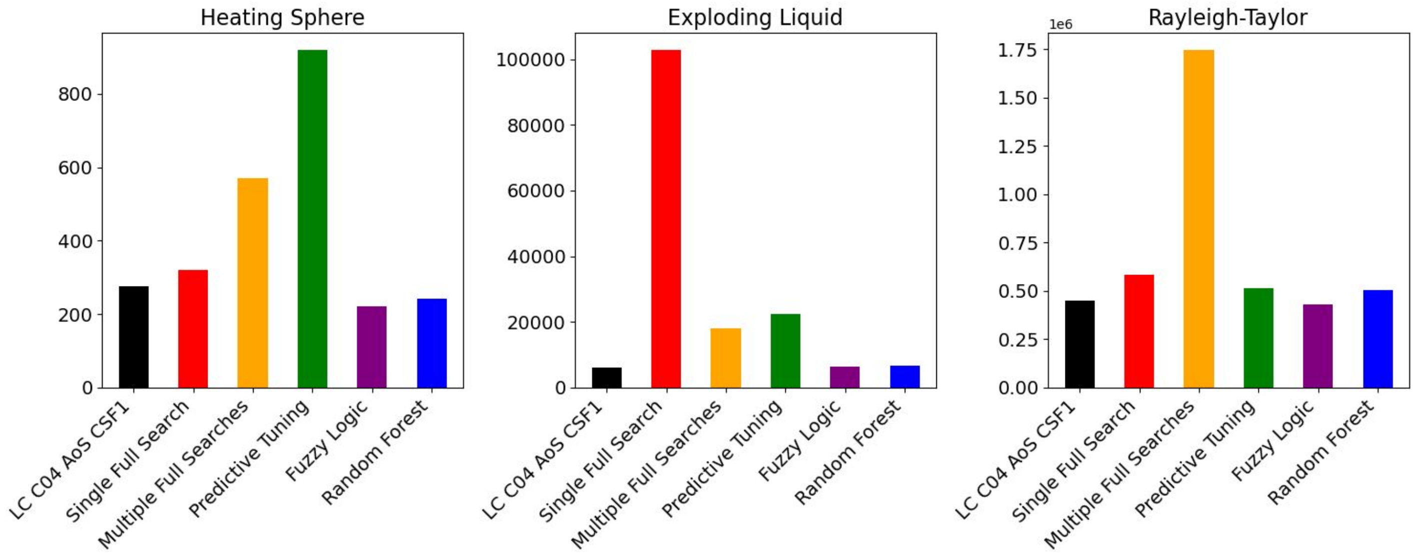

Heating Sphere
0
200
400
600
800
LC C04 AoS CSF1
Single Full Search
Multiple Full Searches
Predictive Tuning
Fuzzy Logic
Random Forest


Exploding Liquid
0
20000
40000
60000
80000
100000
LC C04 AoS CSF1
Single Full Search
Multiple Full Searches
Predictive Tuning
Fuzzy Logic
Random Forest


Rayleigh-Taylor
1e6
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
LC C04 AoS CSF1
Single Full Search
Multiple Full Searches
Predictive Tuning
Fuzzy Logic
Random Forest

Technical University of Munich

# Results



- Expert-Knowledge Fuzzy Logic best.
- Not necessarily any better than picking a single best configuration.
- Random Forests not far behind, and generally more user-friendly.
- Predictive Tuning is the worst.

## Conclusion & Future Work

- We have a portfolio of tuning strategies to serve a range of different effort levels of users.
- But overall, these results suggest a ML-driven approach for future development would be best:
    - Much less effort than expert-knowledge
    - Not much worse performance.
    - No reason why similar or better performance could be achieved with more development or data.
- Coming up with suitable training data depends on use case => Online Learning?