

## Seminar Final Presentation

# Algorithm Selection and Auto-Tuning in AutoPas

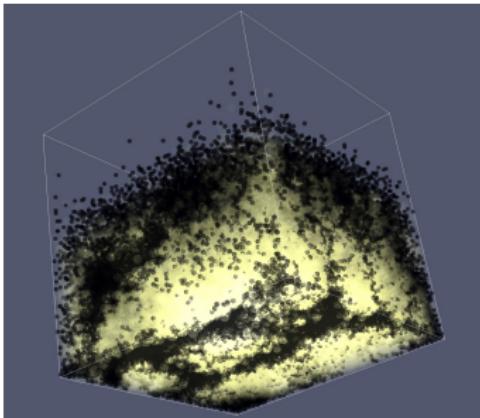
**Manuel Lerchner**  
[manuel.lerchner@tum.de](mailto:manuel.lerchner@tum.de)

## Table of Contents

- 1 Introduction
- 2 Algorithmic Trade-offs
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

## Software Challenges in Molecular Dynamics

- Enormous numbers of particles
  - naively:  $\mathcal{O}(N^2)$  interactions
- Solution: highly optimized algorithms
  - Make full use of hardware
  - Smart data structures
- Every millisecond counts

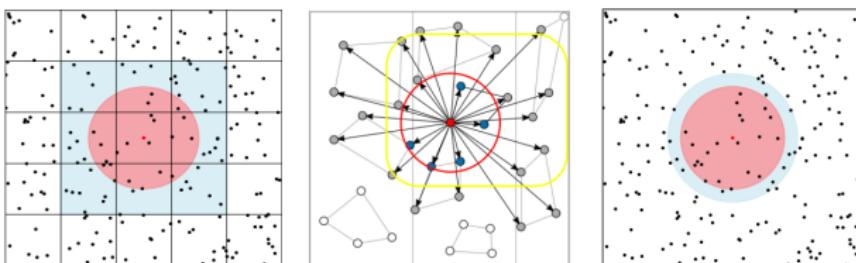


## Table of Contents

- 1 Introduction
- 2 Algorithmic Trade-offs
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

## Trade-off: Particle Containers

- How to identify interacting particles?
- Short-range forces allow for cutoff radius  $\rightarrow \mathcal{O}(N)$  interactions
- Different implementations possible



Linked Cells

Verlet Cluster Lists

Verlet Lists

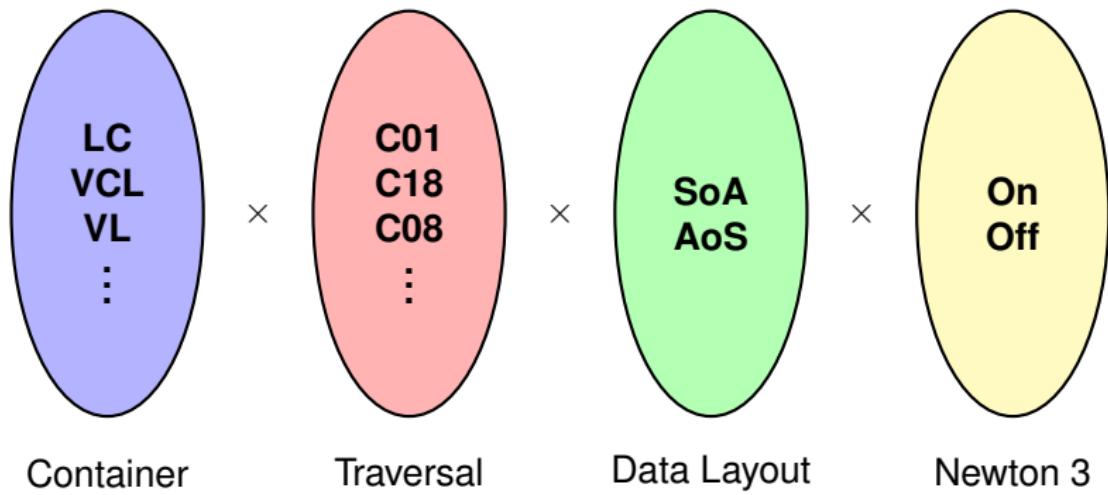
Simpler Memory Access

Lower Memory Overhead

Fewer redundant calculations

## Many more difficult choices

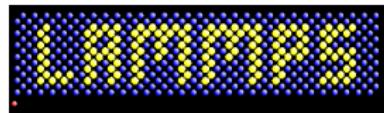
- No silver bullet!
- Hardware- and Scenario-dependent



## Traditional MD Engines



- Based on a single container
- Performance through:
  - Very optimized code
  - Vectorization
- Drawbacks:
  - Suboptimal for some scenarios
  - Require manual tuning



ls1  
Mardyn

## Table of Contents

- 1 Introduction
- 2 Algorithmic Trade-offs
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

# AutoPas

## What is AutoPas?

- Library for arbitrary N-body simulations
- Implements all algorithms
  - Modular
  - All\* configurations possible
- Idea: AutoTuning
  - Finds optimal configuration  
→ Thus optimal performance
- Arbitrary user code

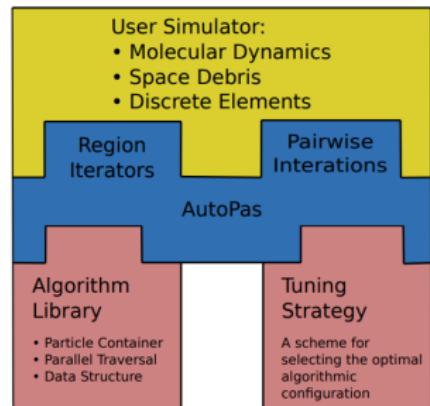


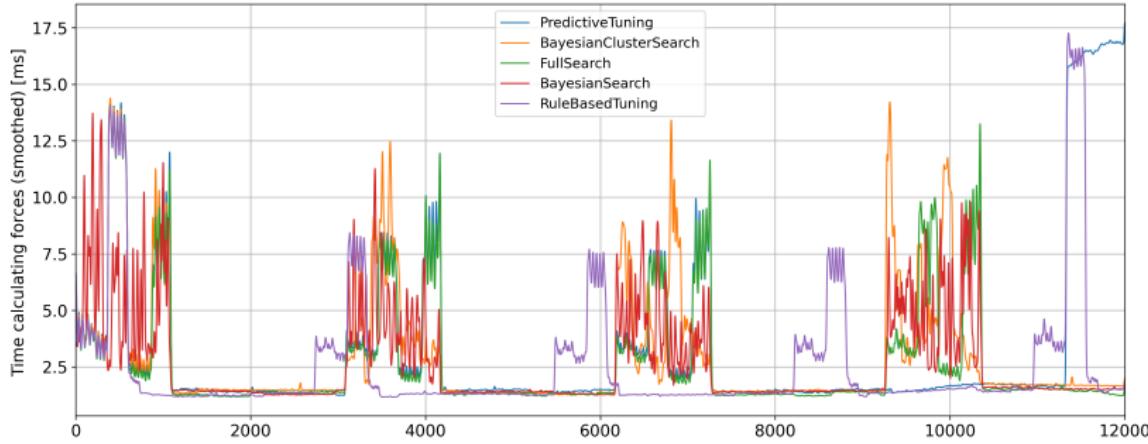
Figure: [Newcome et al., 2023]

## Table of Contents

- 1 Introduction
- 2 Algorithmic Trade-offs
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

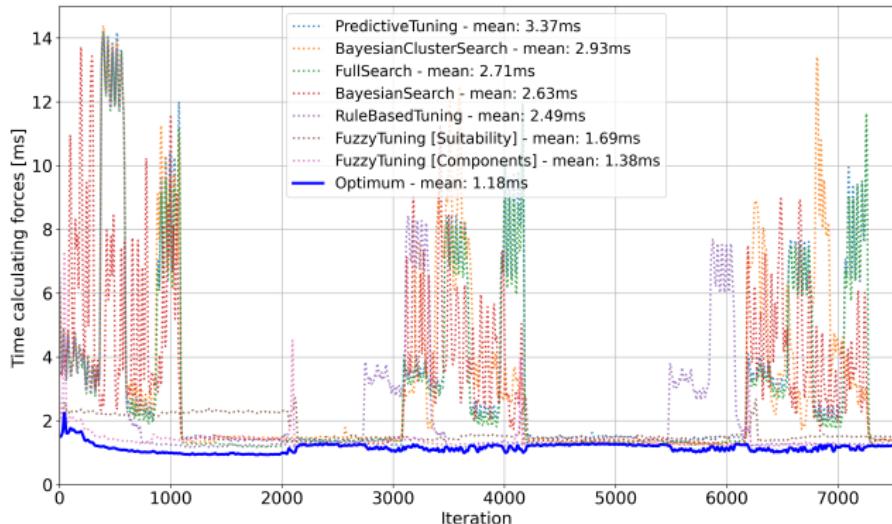
## Auto-Tuning in AutoPas

- Tuning-Phase: Evaluate promising configurations
- Simulation-Phase: Use best configuration
- Capable of reducing simulation time!



## Problems of Auto-Tuning

- Overhead from evaluating suboptimal configurations
  - Orders of magnitude slower!
- Unnecessary periodic re-tuning

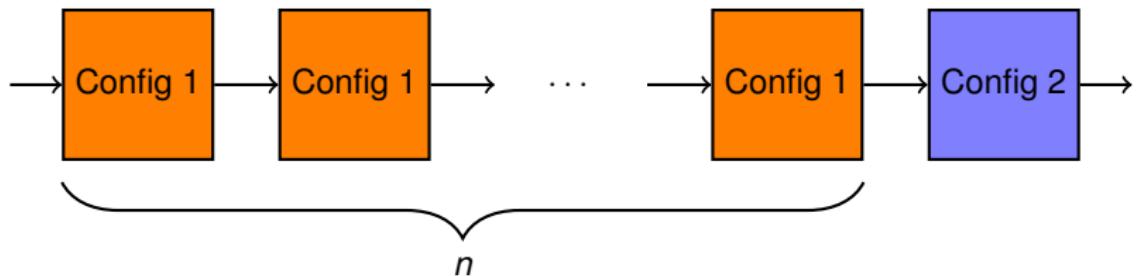


## Table of Contents

- 1 Introduction
- 2 Algorithmic Trade-offs
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

## Early Stopping Optimization

- Idea: Stop long-running evaluations early
- Configurations get sampled multiple times
- Naive approach:
  - Last sample performed poorly?
  - Stop further evaluations

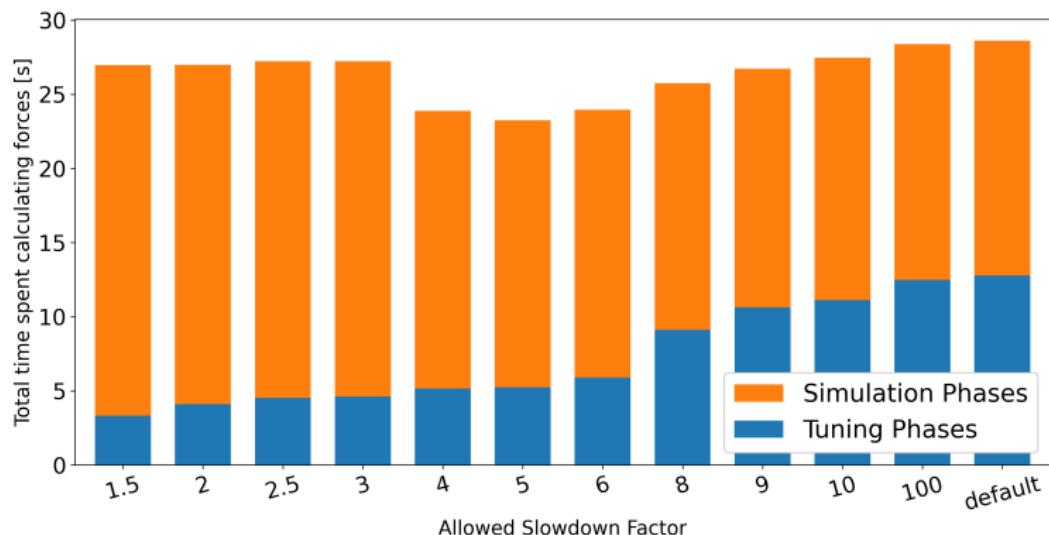


## Naive Early Stopping

- Changes to AutoPas are minimal
  - Keep track of best performance seen so far
  - $\text{slowdownFactor} = \frac{t_{\text{sample}}}{t_{\text{best}}}$
  - Call `retune()` if `slowdownFactor` exceeds threshold
- Hyperparameter: *allowedSlowdownFactor*
- Which *allowedSlowdownFactor* should be used?
  - *allowedSlowdownFactor* → 1: Many aborts
  - *allowedSlowdownFactor* → ∞: No early stopping
- Empirical evaluation needed

## Benchmark: Exploding Liquid + Predictive Tuning

- Optimal at  $allowedSlowdownFactor = 5$
- From 28.6s to 23.2s
  - 18.9% reduction



## Summary and Future Work

- AutoPas Library
  - Flexible and optimal N-body simulations
  - AutoTuning can cause overhead
- (Naive) Early stopping is beneficial
  - Never increased runtime!
  - Potential for further improvements
- *allowedSlowdownFactor* probably not universal
  - More benchmarks needed!
  - Investigate other stopping criteria

**Thank you for your attention!**

**Questions?**

## References I

-  Newcome, S. J., Gratl, F. A., Muehlhaeusser, M., Neumann, P., and Bungartz, H.-J. (2024). Autopas: Dynamic algorithm selection in molecular dynamics for optimal time and energy.  
*In SIAM Conference on Parallel Processing (PP24)*. SIAM.
-  Newcome, S. J., Gratl, F. A., Neumann, P., and Bungartz, H.-J. (2023). Towards the smarter tuning of molecular dynamics simulations.  
Amsterdam, The Netherlands.