Online Model Swapping for Architectural Simulation

Patrick Lavin, Jonathan Beard

HotCSE
21 October 2020

Outline

Background

- Motivation
- Research Idea
- Research Goal

Online Model Swapping

- Phase Analysis
- Cache Models
- Model Selection
- Model Swapping

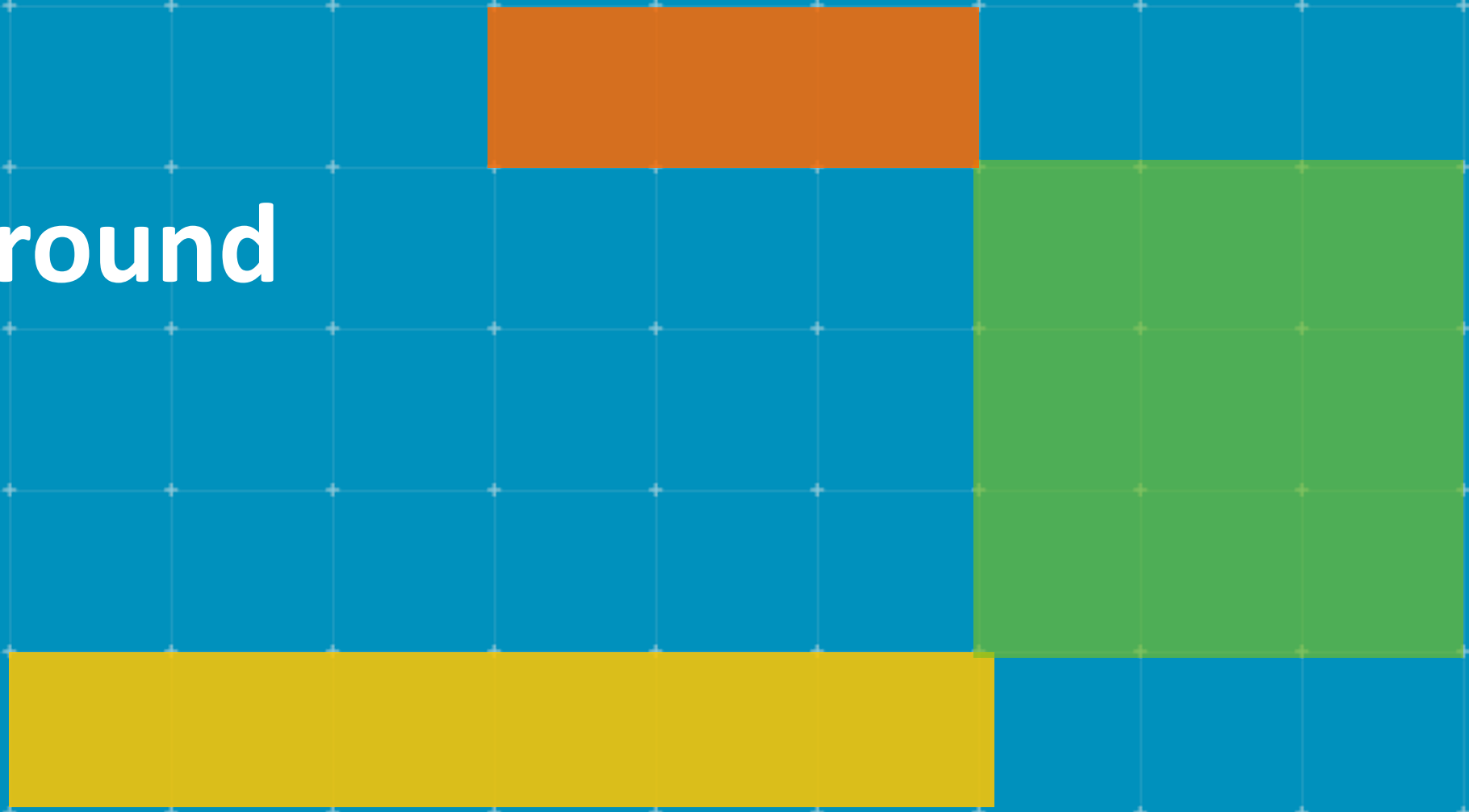
Results

- Methodology (Meabo and SVE-Cachesim)
- Accuracy
- Locality
- Complexity

Final Remarks

- Future work

Background



arm

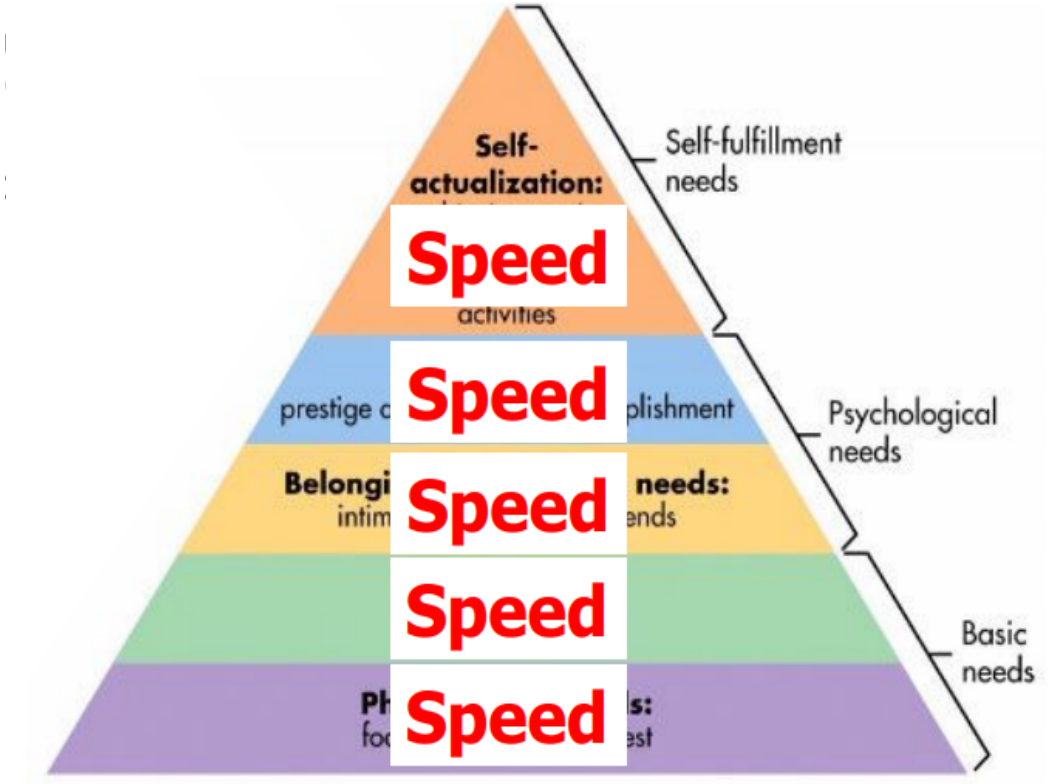
Motivation

Speed, of course

1. Detailed architectural simulation takes a long time and will only take longer as chips get more complicated.

2. Detail isn't often needed for every part of an application.

3. If we could automatically select simpler, faster models for parts of an application, we might be able to speed up simulation.



Maslow's Hierarchy, revisited

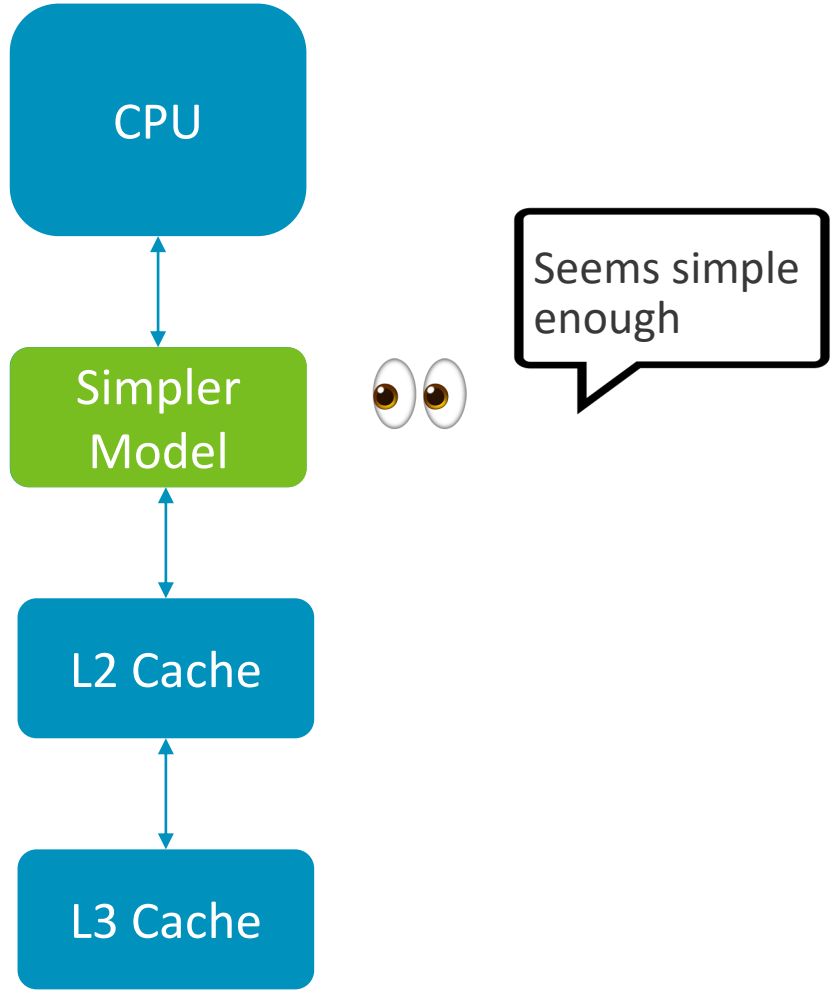
Credit: Onur Mutlu, https://slideplayer.com/slide/17641291/

arm

Research Idea

Online Model Swapping

- We'll examine the behavior of a component of the simulation
- If some parts of the program seem simple enough to predict, we can swap in a simpler one
- Hopefully this doesn't break the rest of the simulation



arm

Research Output

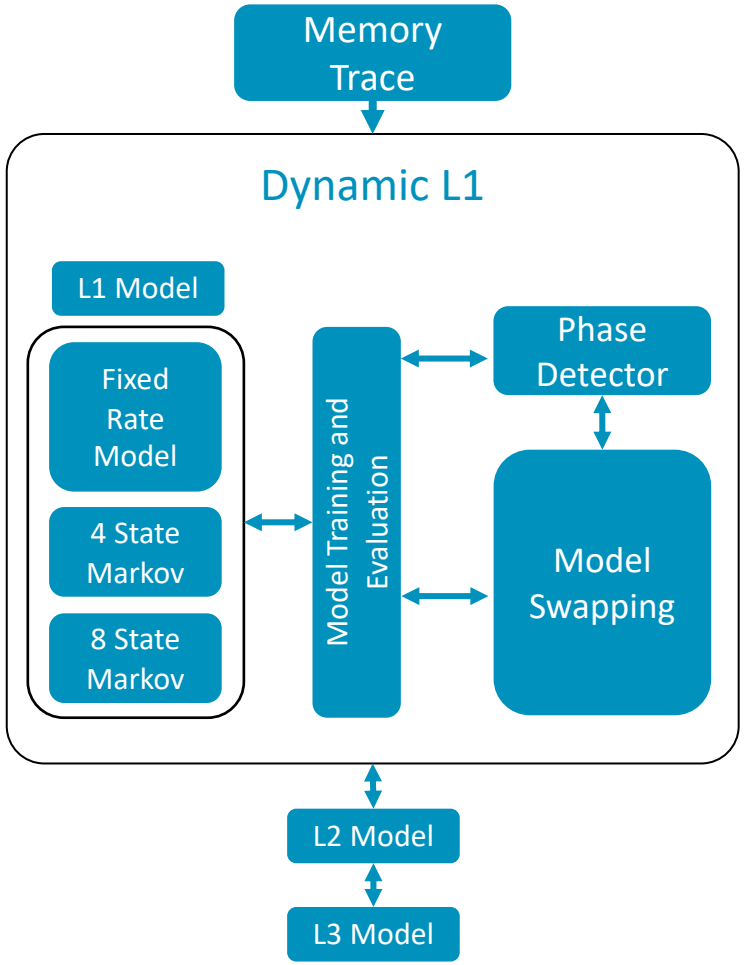
Online Model Swapping

Key Takeaways

1. Online model swapping works.
2. This work provides a framework for future research in online model swapping.

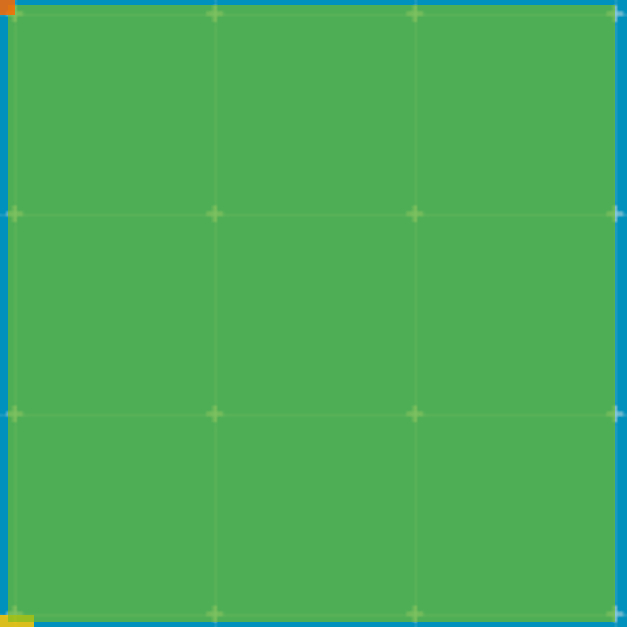
Results

1. Our system trains several statistical cache models during simulation.
2. We score them and choose the best to swap in, in place of the detailed L1 cache model.
3. We do this with only an 8% error in the simulated cycle count, while running our simpler models for over 90% of the simulation .



arm

Online Model Swapping



arm

Sub-problems to Tackle

- When do we evaluate models? How can we simplify the problem of training the models?
- What statistical models can we use to replace the base model?
- How do we choose between models?
- How do we swap out models?

Phase Detection

Alternate Models

Model Selection

Model Swapping

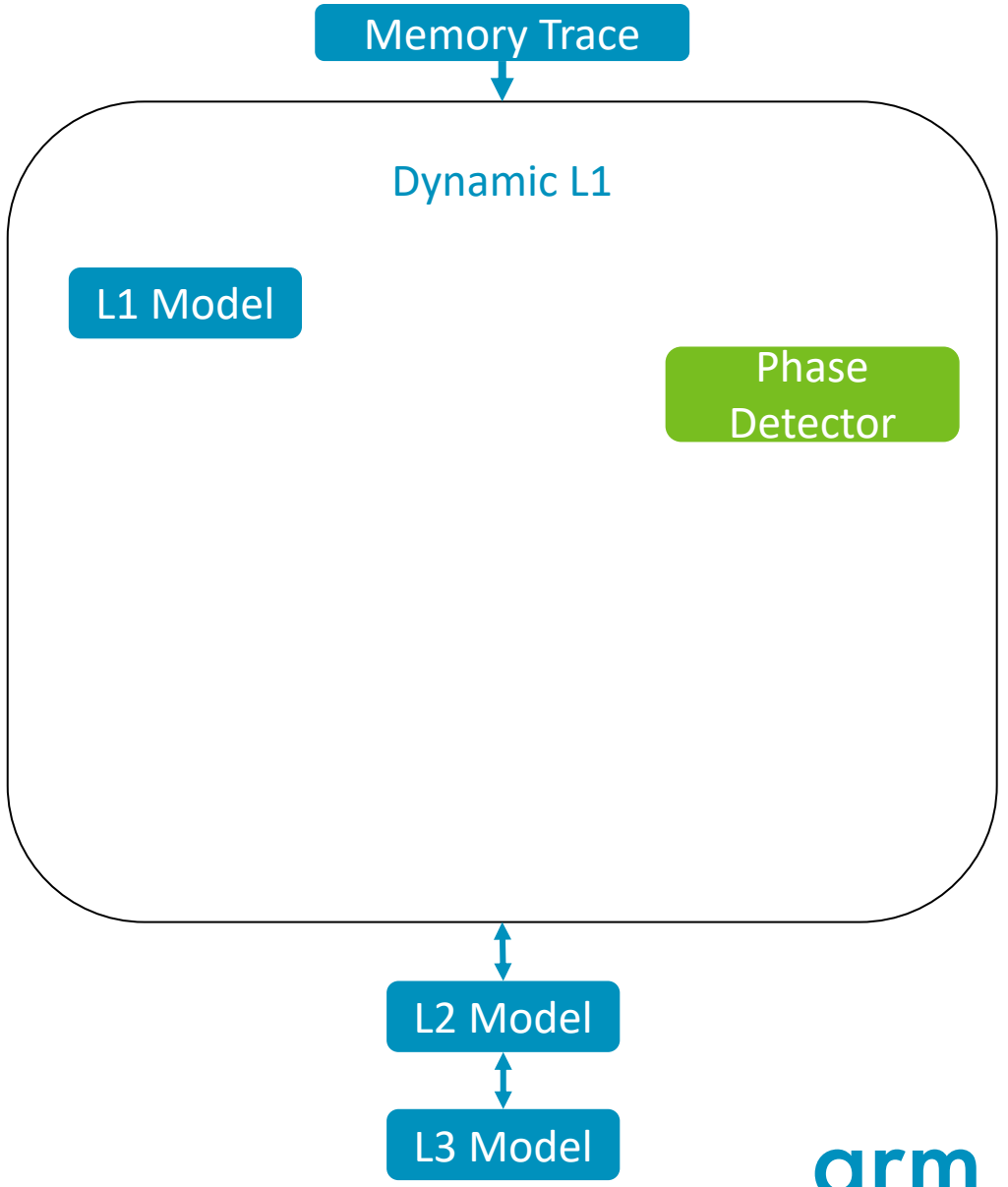
arm

Phase Detection

- When do we evaluate models?
- How can we simplify the problem of training the models?

Working Set-Based Phase Detection

- The working set is the set of recently used instruction pointers.
- Form a working set signature by hashing the instruction pointers in an interval into a bit vector.
- If the signature is close to one already encountered, classify that interval as being part of the same phase.
- Dhodapkar, Smith ISCA'02



arm

Alternate Cache Models

- What statistical models can we use to replace the base model?

Fixed Rate

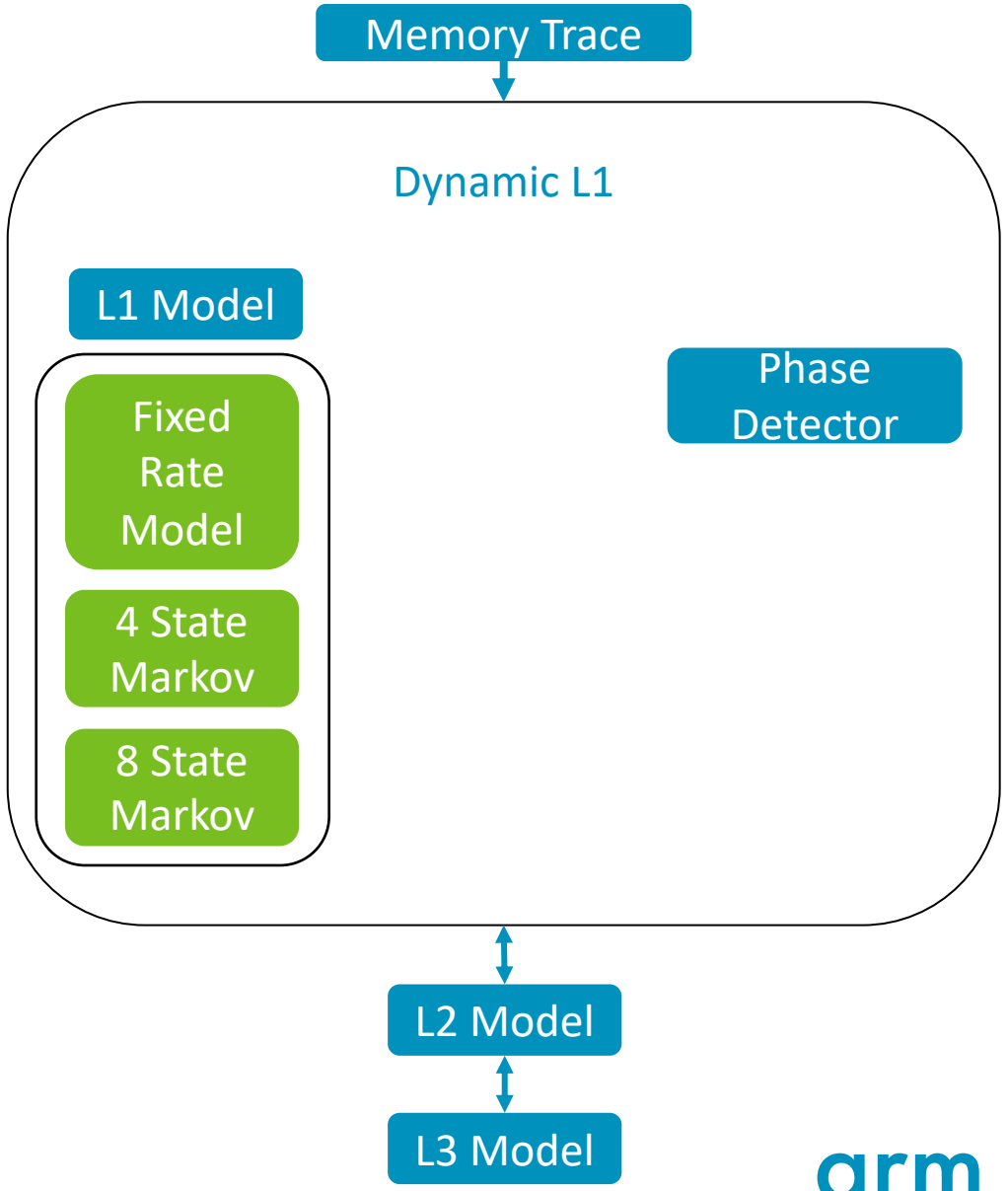
- Learn the hit rate of a phase, and use only that for prediction.

4-State Markov

- Learn the transition matrix for the states *ReadHit*, *ReadMiss*, *WriteHit*, *WriteMiss.*
- Use both the previous state and the current request for prediction.

8-State Markov

- Add *Near* and *Far* versions of the above states.
- *Near* means the access is on the same cache-line as the previous access.
- *Far* is anything else.



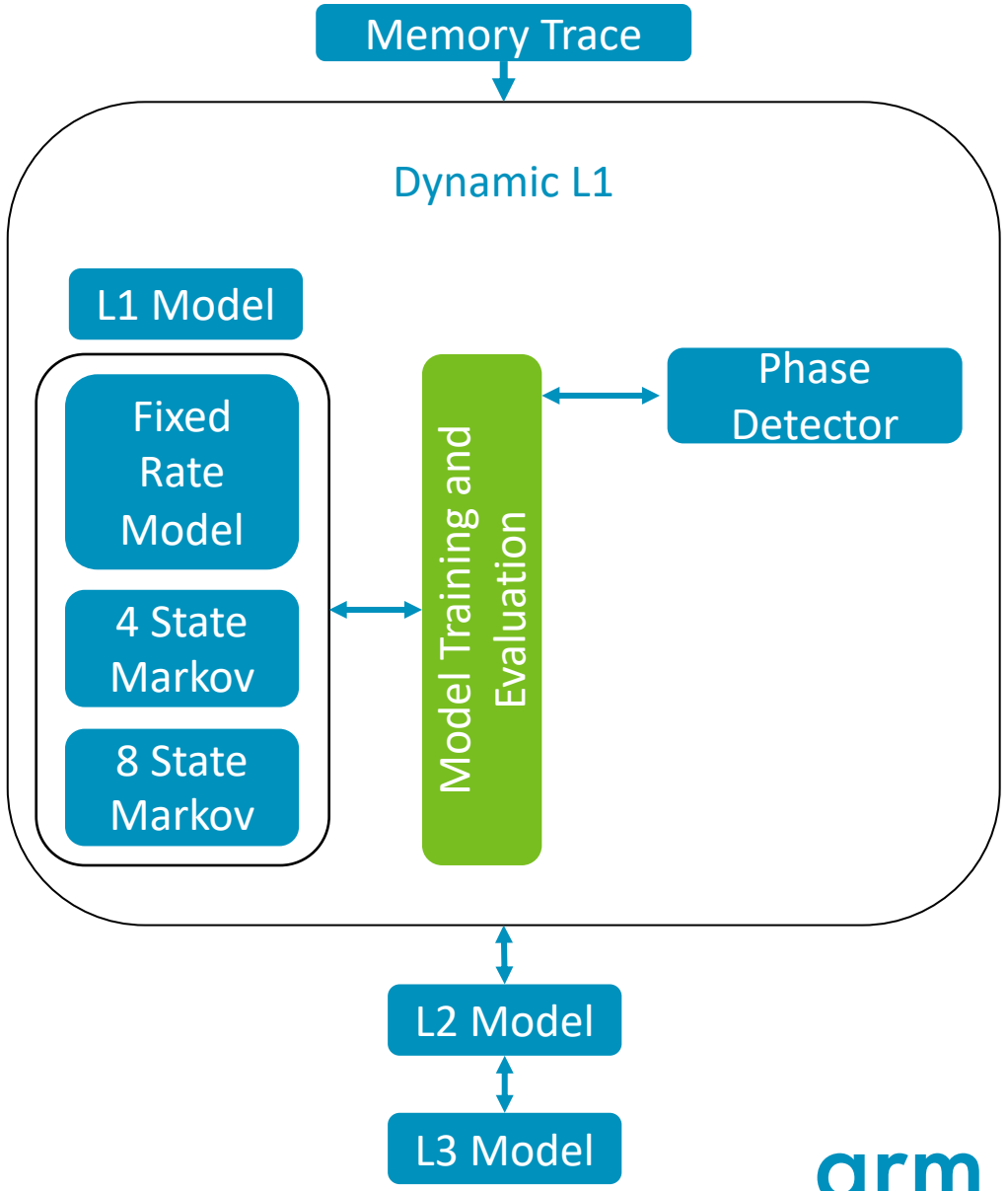
arm

Model Selection

- How do we choose between models?

Scoring Criteria

1. **Accuracy**
    1. How well does the partially trained model predict hits and misses?
2. **Near percentage for misses**
    1. Proxy for spatial locality
3. **Model State Size** (percent of base cache)
4. **Model Complexity** (percent of base cache)



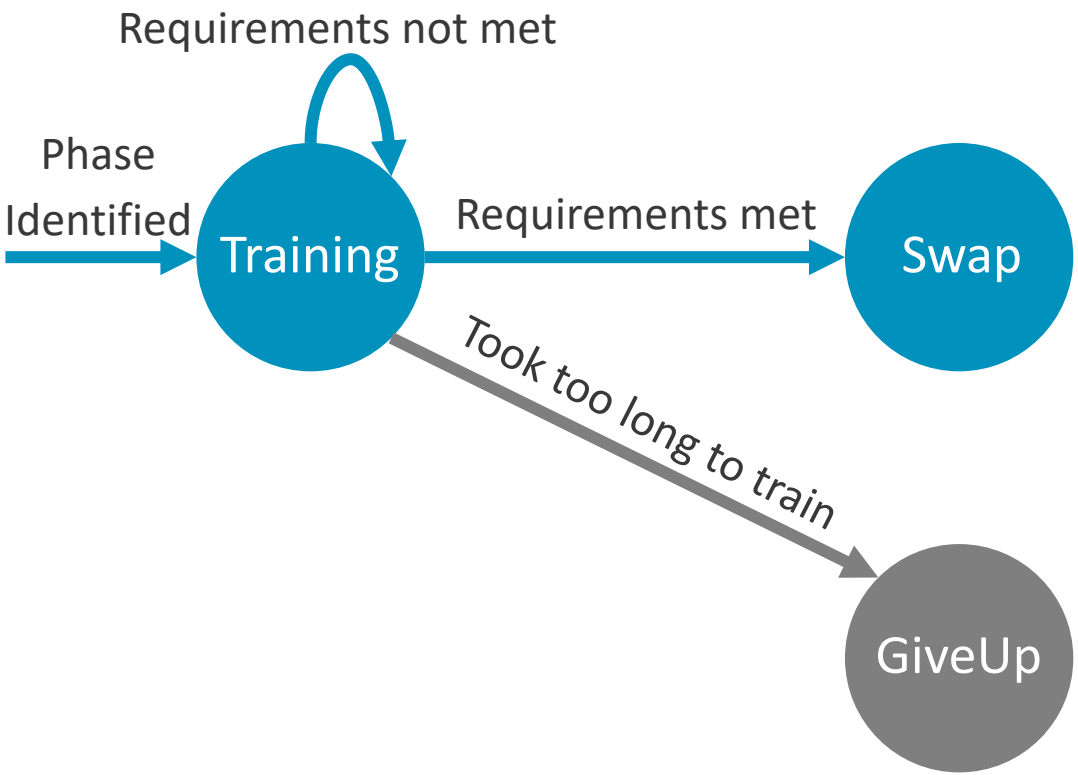
arm

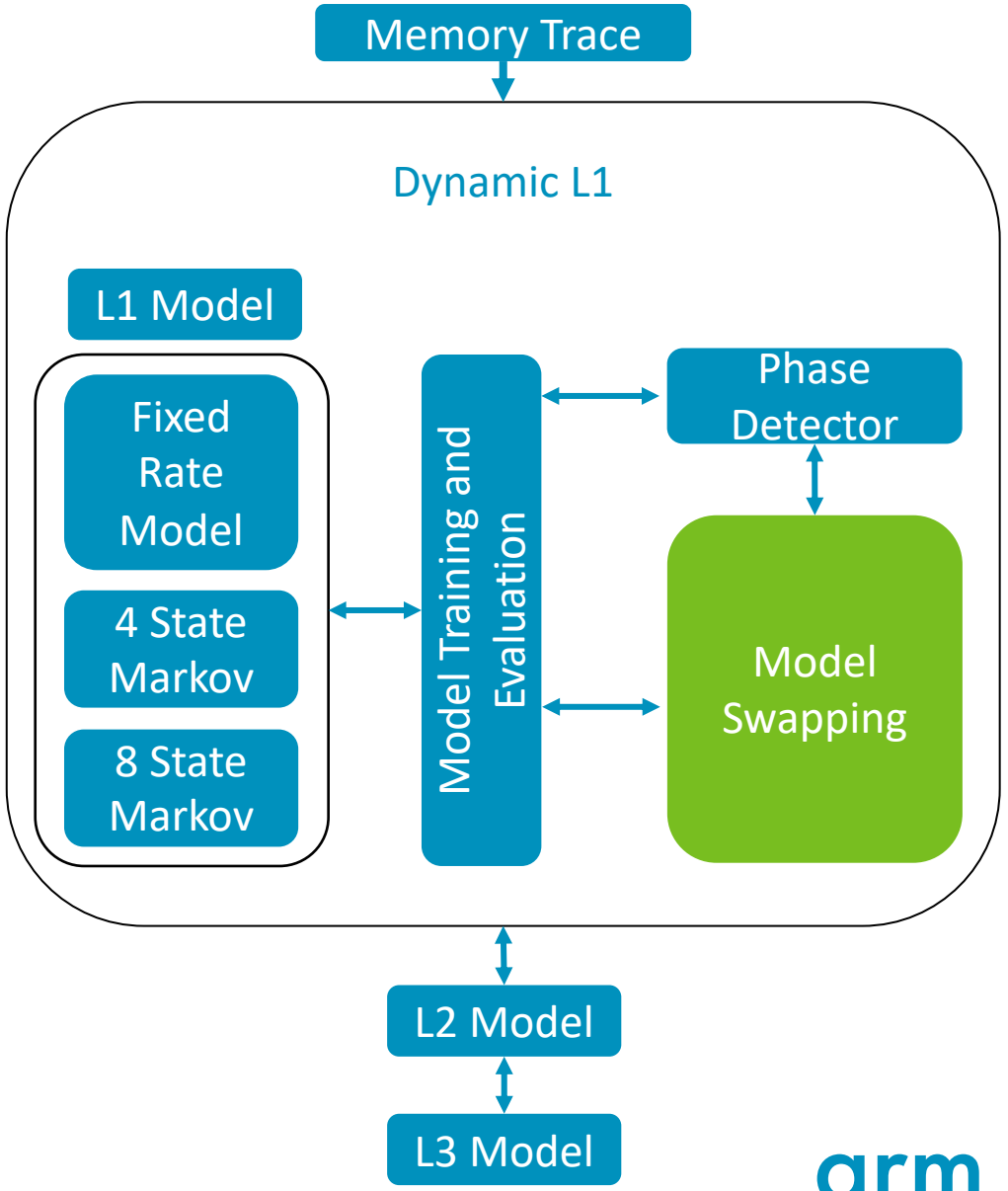
Model Swapping Algorithm

- How do we swap out models?

The Model Swapping Algorithm

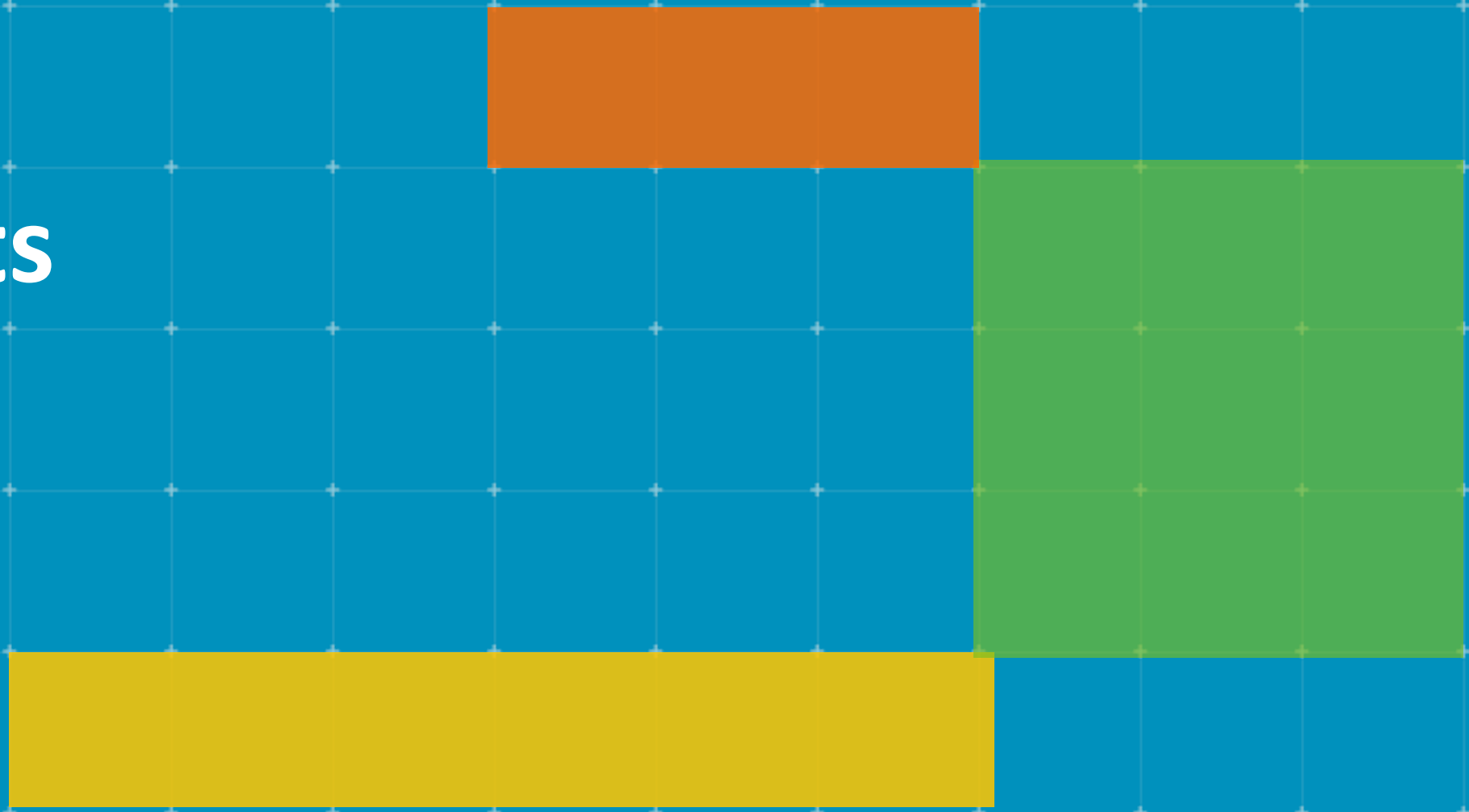
- Run for every phase





arm

Results



arm

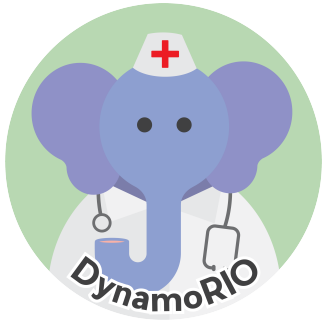
Methodology

Trace generation and simulation infrastructure

DynamoRIO

memtrace grabs all memory references

- Single-threaded trace
- Output:
  - [Instruction Ptr., Virtual Addr., Read/Write]



SVE-Cachesim

Simple, open source, cache simulator written in Python

- Takes DR trace an input
- In-order simulation, no outstanding requests
- 3-levels (configurable, here's ours):
  - L1: 32KiB, 8-way associative
  - L2: 256KiB, 8-way associative
  - L3: 1MiB, 32-way associative
- We extend *sve-cachesim* with our phase analysis and model swapping functionality

arm

Maebo Benchmark

What it is

- In their own words: “Meabo is a multi-phased multi-purpose micro-benchmark.”
- Its distinct phases make it a good test of our phase analysis tool, and make our results easier to interpret

Selected phases

- Floating-point & integer computations with good data locality
- Vector addition
- Random memory accesses

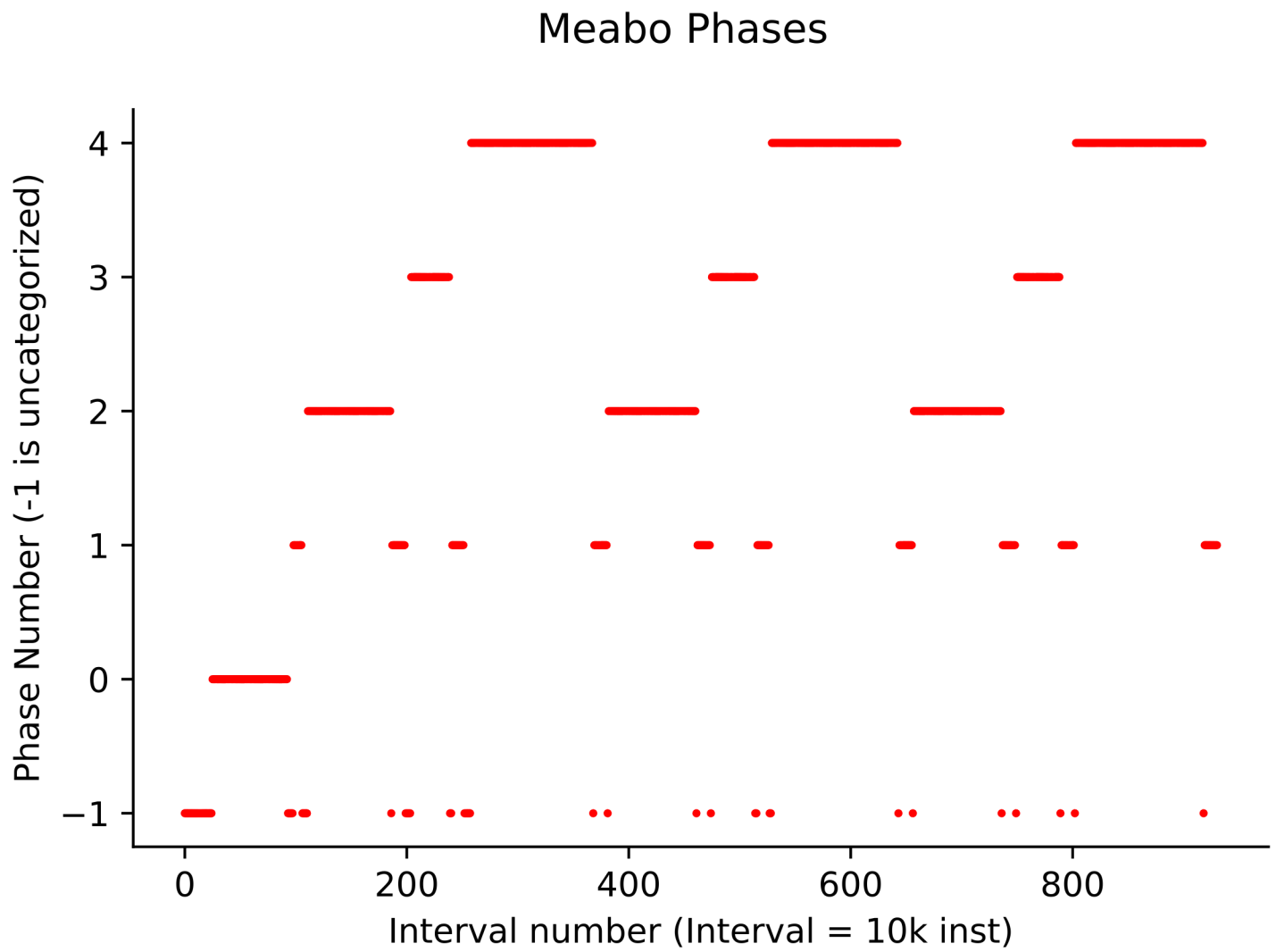
Changes

- Added a marker phase between phases
- Added a loop to run each phase 3 times, instead of just once

Phase Analysis

Phase Key

- -1: uncategorized
- 0: Initialization
- 1: Marker phase
- 2: High locality
- 3: Vector add
- 4: Random



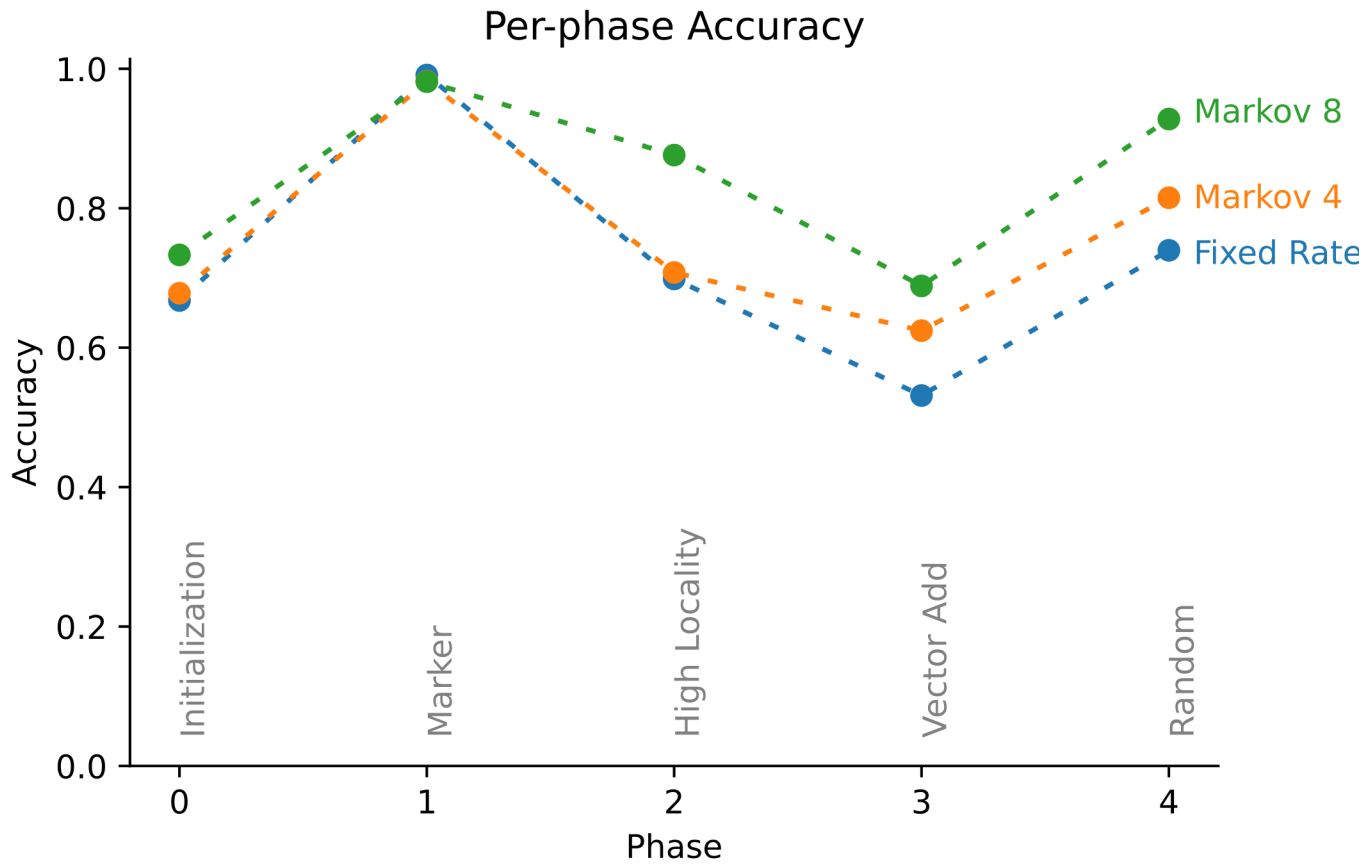
arm

Accuracy

Comparing Hits

We train the model for 2 intervals, then swap in the trained model

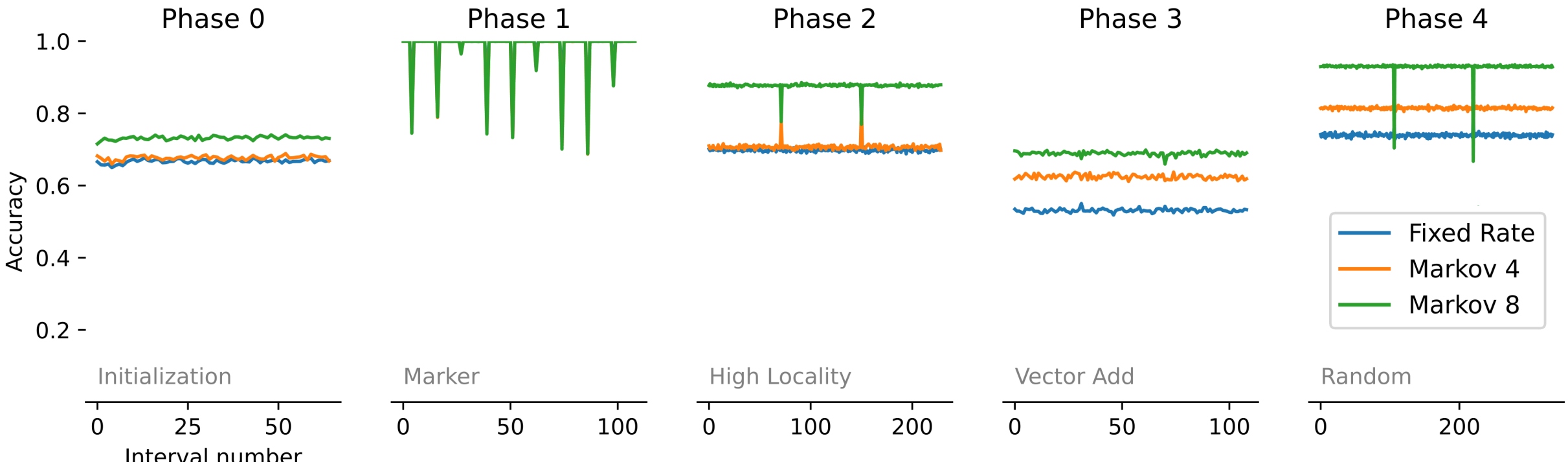
- From inspection, the model parameters do not change much after the second training interval
- *Accuracy* is the percentage of accesses correctly predicted



arm

Accuracy Over Time

- Does accuracy vary during a phase?
- Does it get worse over time?



arm

Locality

Let's not break the rest of the simulation

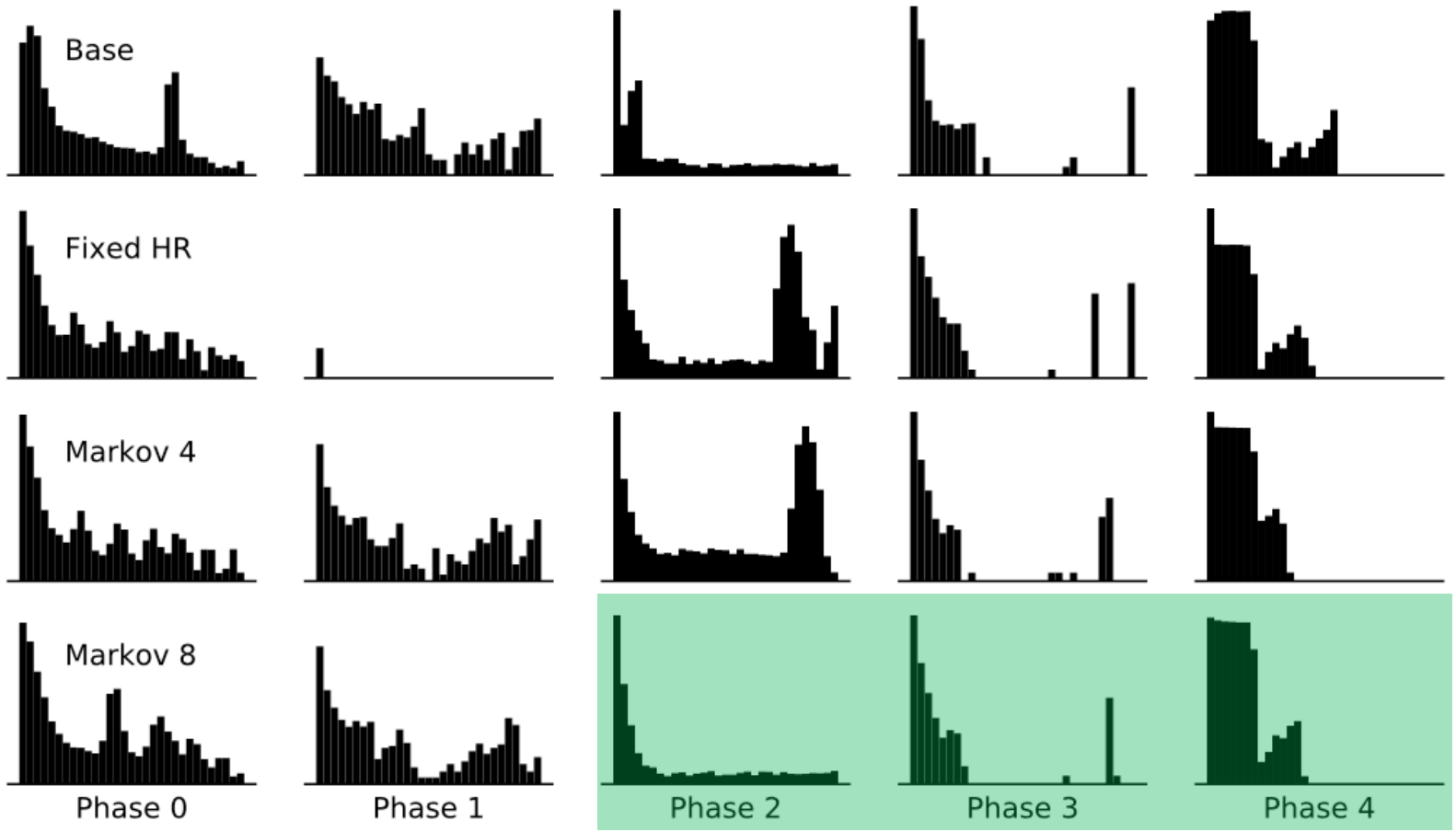
- Percent change in hit rate for each cache level and total cycles
- Absolute count displayed for base, percent change for the rest
- It seems we may be messing up the locality properties of the L1 misses with some of these models…

Accuracy	L1 Hits	L2 Hits	L3 Hits	Total Cycles
Base	$7.7x10^7$	$7.8x10^5$	$2.5x10^5$	$1.4x10^8$
Fixed Rate	-0.08%	54.26%	-71.13%	-27.89%
Markov 4	-0.37%	46.14%	-52.50%	-23.06%
Markov 8	-0.19%	12.27%	-4.71%	-7.57%

Locality

Reuse distance histogram of the addresses entering the L2 cache

- Cache-line granularity
- Truncated at distance=200



arm

Model Selection

Putting it all together

Scoring

Our framework is able to train all 3 models simultaneousl,y and score them all using the previously mentioned distance metric

- Accuracy, Near count, Model size, Model complexity
- The base model will have a score of sqrt(2)=1.41

Scoring	Phase 0	1	2	3	4
Fixed Rate	1.07	.06	1.15	1.01	1.12
Markov 4	1.03	1.0	1.16	.90	1.07
Markov 8	.60	.22	.89	.68	1.03

Accuracy	L1 Hits	L2 Hits	L3 Hits	Total Cycles
Base	$7.7x10^7$	$7.8x10^5$	$2.5x10^5$	$1.4x10^8$
Fixed Rate	-0.08%	54.26%	-71.13%	-27.89%
Markov 4	-0.37%	46.14%	-52.50%	-23.06%
Markov 8	-0.19%	12.27%	-4.71%	-7.57%
All	0.07%	10.15%	-4.80%	-8.00%

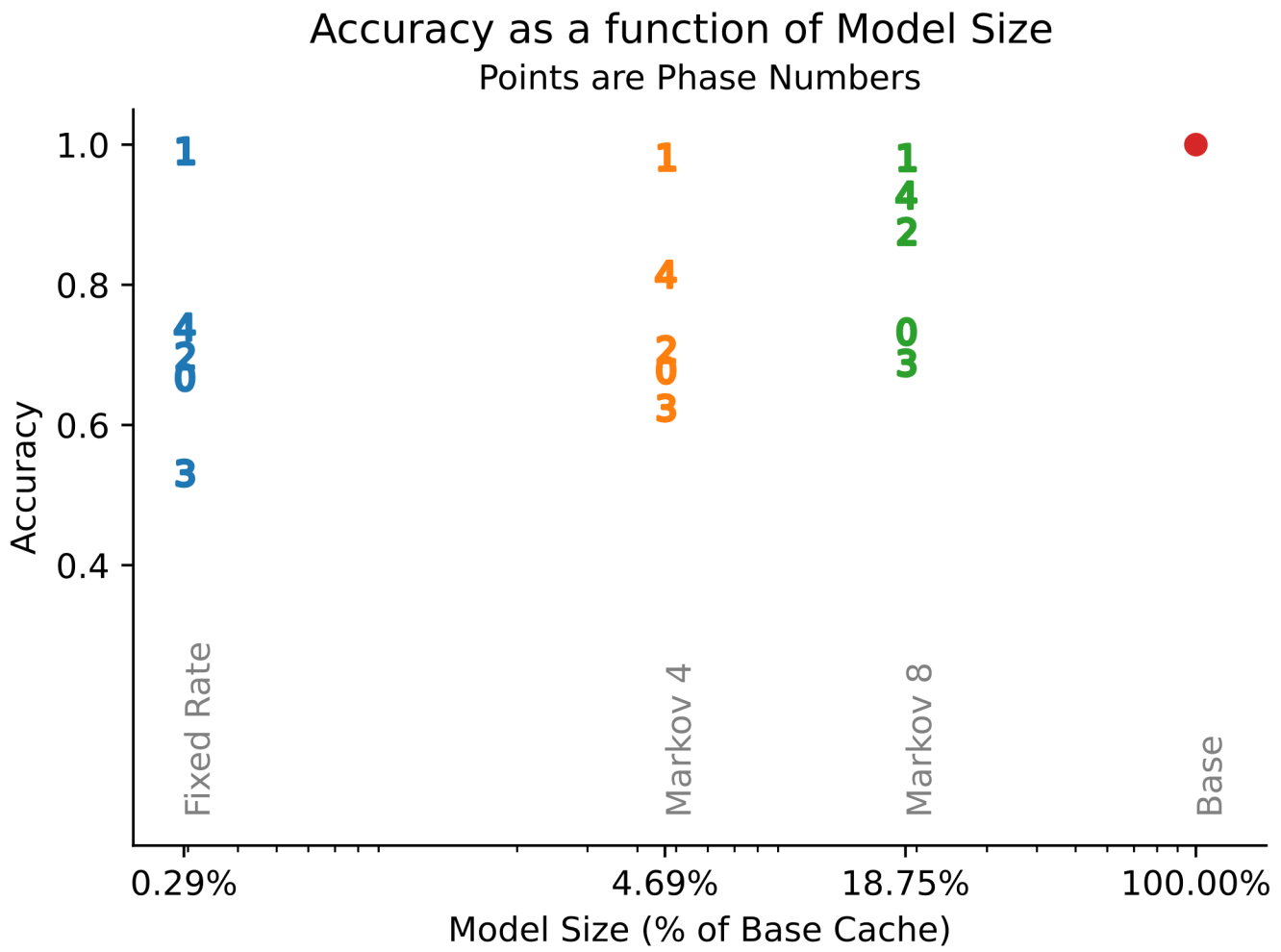
Final Remarks

arm

Future Work

Lots To Do!

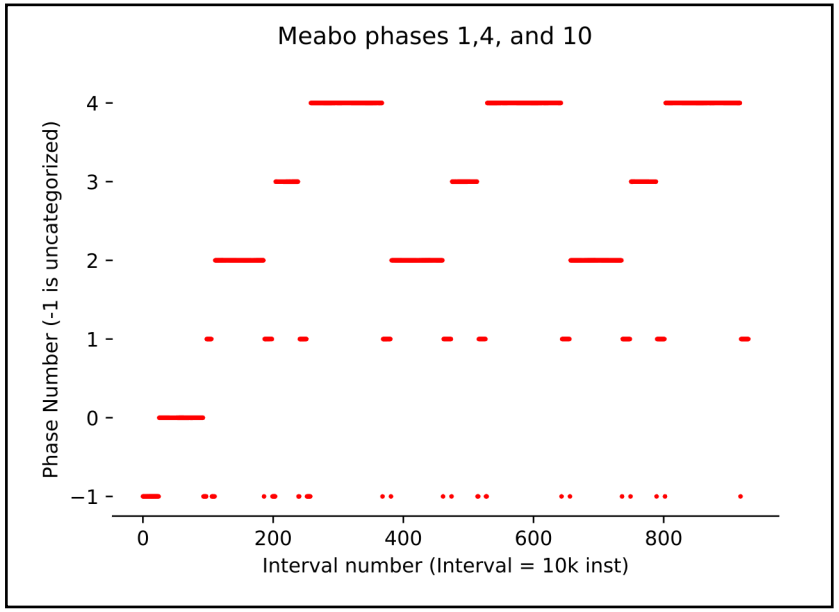
- More cache models
  - LSTM
- Integrate into SST
- Further exploration
  - Model selection criteria
  - Phase analysis dynamic interval size
  - Model re-evaluation
  - Un-swapping methodology
- More components

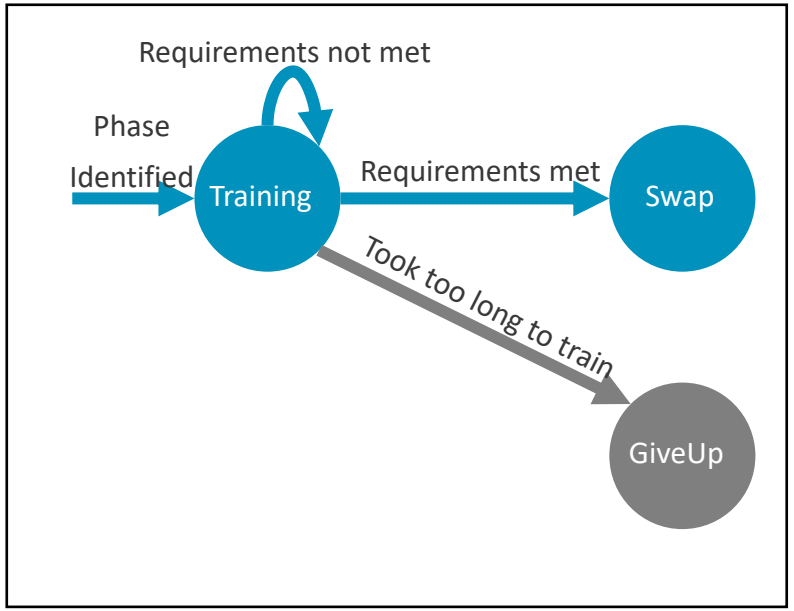


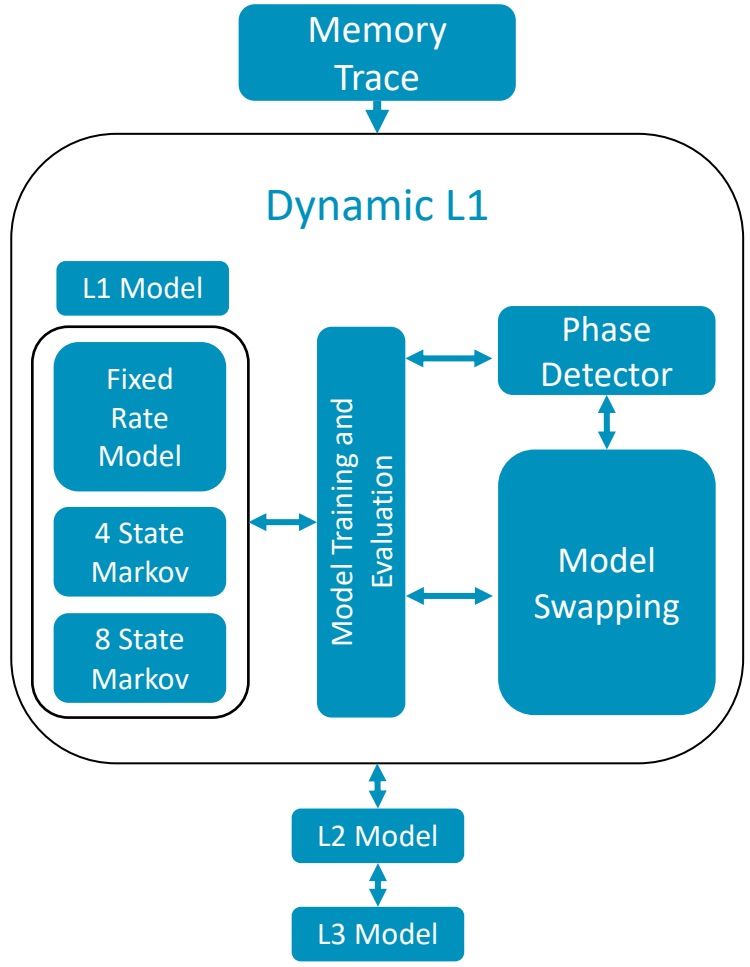
arm

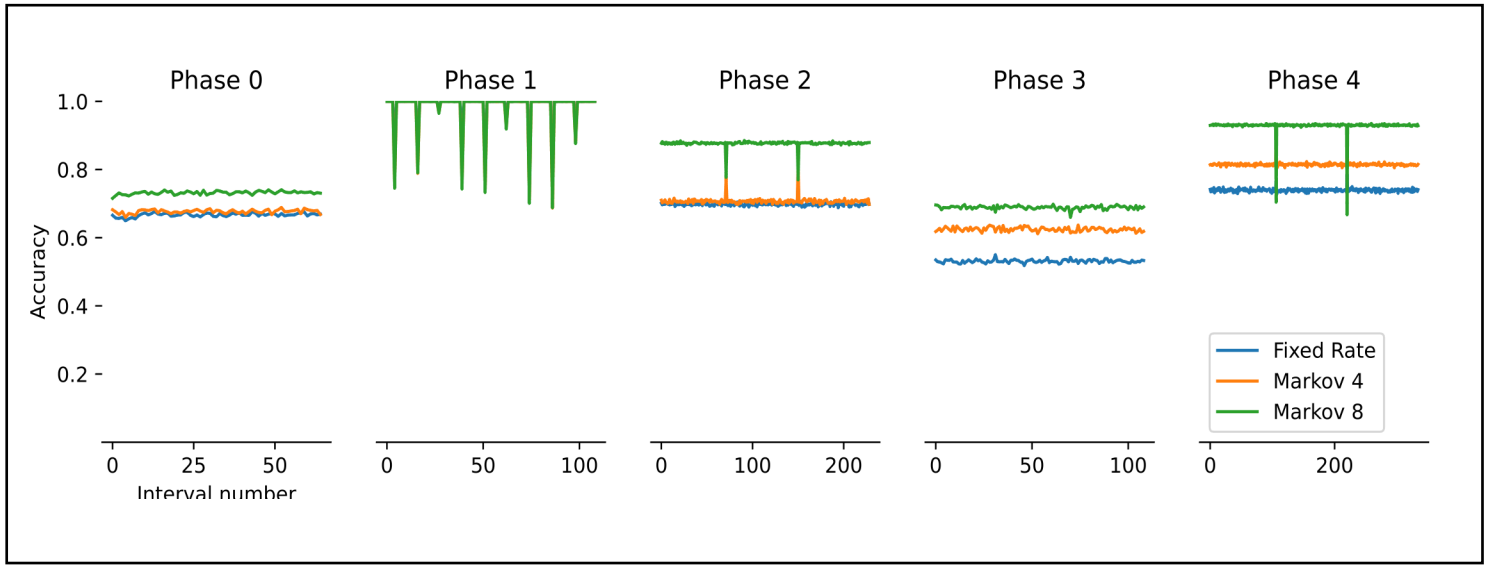
Thank You!
Danke!
Merci!
谢谢!
ありがとう!
Gracias!
Kiitos!

Online Model Swapping for Architectural Simulation









arm