# The Results of SAT Competition 2025

Cayden Codel, Katalin Fazekas, **Marijn Heule**, Markus Iser

SAT 2025 Conference, Glasgow, Scotland

August 14, 2025

# Competition Overview

## SAT Competitions

- 3 competitions in the 90s (1992,1993, 1996)
- 17 SAT Competitions (2002–)
- 5 SAT Races (2006, 2008, 2010, 2015, 2019)
- 1 SAT Challenge (2012)

## Goals

- Promotion of SAT solvers and their development
- Compilation of new challenging benchmarks
- Evaluation of current state-of-the-art solvers

# Key rules

- Certified results of unsatisfiability using proof logging
  - Instance is "not solved" if proof checker finds times out
- Disqualification of buggy solvers
  - Producing an incorrect model
  - Report UNSAT on a known satisfiable instance
- Mandatory solver descriptions + open source
- Ranking scheme: PAR-2
  - Favors solvers that are faster (not only count solved instances)
- BYOB (Bring Your Own Benchmarks)
  - At most 20 instances per participant are used

# New This Year

Two new organizers:

- Cayden Codel and Katalin Fazekas
- Thanks to Matti Jarvisalo and Martin Suda!

# New This Year

Two new organizers:

- Cayden Codel and Katalin Fazekas
- Thanks to Matti Jarvisalo and Martin Suda!

New infrastructure for the sequential track:

- Dirk Beyer's cluster at LMU
- Thanks to StarExec!

# New This Year

Two new organizers:

- Cayden Codel and Katalin Fazekas
- Thanks to Matti Jarvisalo and Martin Suda!

New infrastructure for the sequential track:

- Dirk Beyer's cluster at LMU
- Thanks to StarExec!

Parallel track timeout: 1000s (previously 5000s)

Skip Cloud track: too demanding for AWS organizers

# Competition Summary

Main Track: 400 benchmarks

- 184 new submissions (73 SAT, 89 UNS, 22 UNK)
- 116 unused old submissions (38 SAT, 38 UNS, 40 UNK)
- 100 Anniversary track submissions (58 SAT, 42 UNS)
- 26 sequential solvers
- 8 parallel solvers

# Multiple Verified Checkers

Participants picked one of these options:

- Verified LRAT and LPR Proof Checking with cake_lpr
  Yong Kiam Tan, Marijn J. H. Heule, and Magnus O. Myreen
- GRAT: a formally verified (UN)SAT proof checker
  Peter Lammich
- VeriPB and CakePB: Verified Pseudo-Boolean Proofs
  by Markus Anders, Bart Bogaerts, Benjamin Bogø, Arthur Gontier, Wietze Koops, Ciaran McCreesh, Magnus O. Myreen, Jakob Nordström, Adrian Rebola-Pardo, Andy Oertel, and Yong Kiam Tan

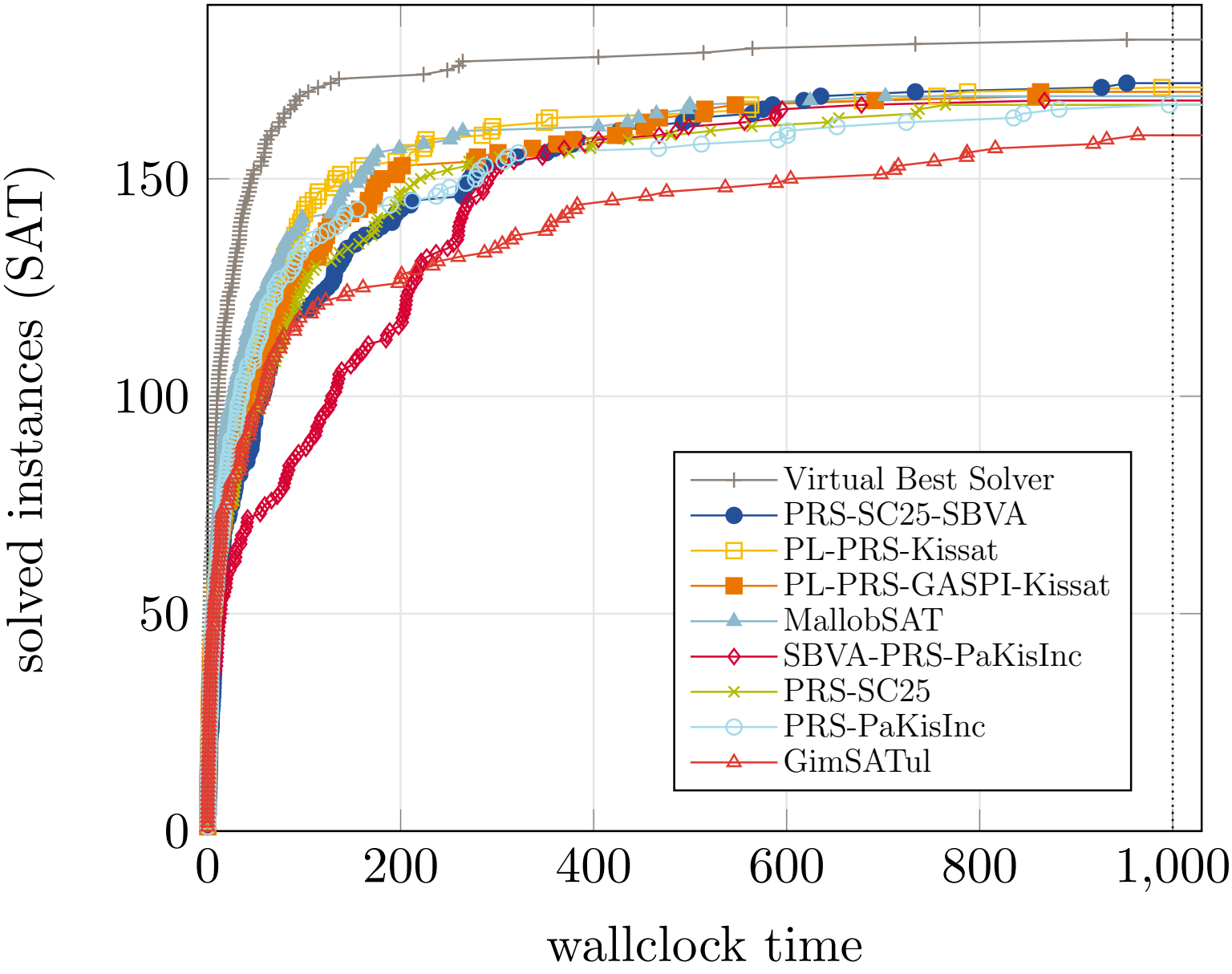
Timeout:

- Solver: 5000 seconds
- Checker tool chain: 45000 seconds

# Parallel Track SAT

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | PL-PRS-Kissat | Mazigh Saoudi, Souheib Baarir, Julien Sopena, Thibault Lejemble, and Sabrine Saouli | 191.21 | 171 |
| 2 | MallobSat | Dominik Schreiber, Niccolo Rigi-Luperti, and Armin Biere | 201.04 | 169 |
| 3 | PRS-SC25-SBVA | Yuhang Qian, Zhihan Chen, Xindi Zhang, and Shaowei Cai | 214.26 | 172 |

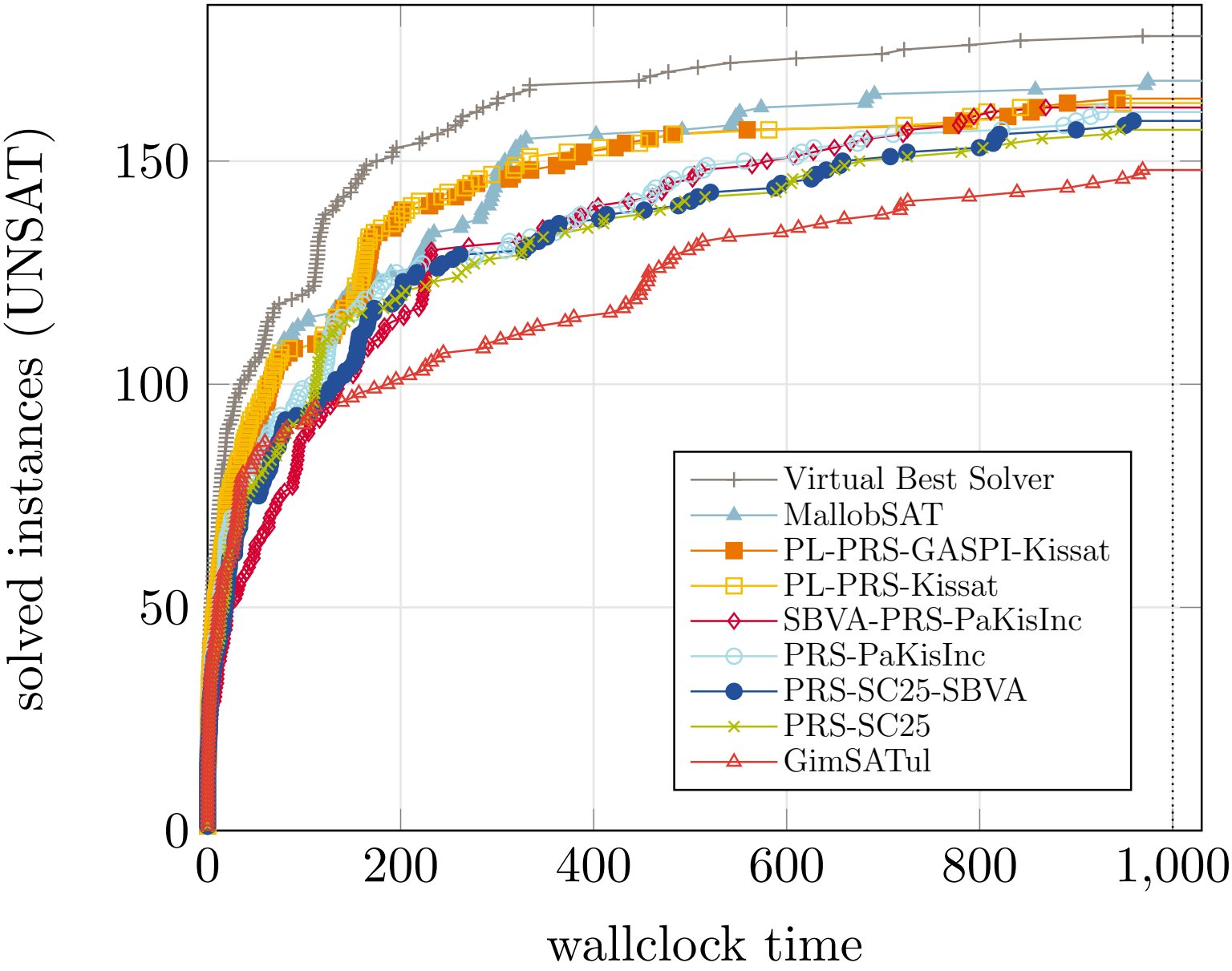
# Parallel Track SAT Plot



# Parallel Track UNSAT

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | MallobSat | Dominik Schreiber, Niccolo Rigi-Luperti, and Armin Biere | 231.65 | 168 |
| 2 | PL-PRS-GASPI-Kissat | Mazigh Saoudi, Souheib Baarir, Julien Sopena, Thibault Lejemble, and Sabrine Saouli | 266.85 | 164 |
| 3 | PRS-PaKisInc | Rodrigue Konan Tchinda and Clementin Tayou Djamegni | 327.21 | 161 |

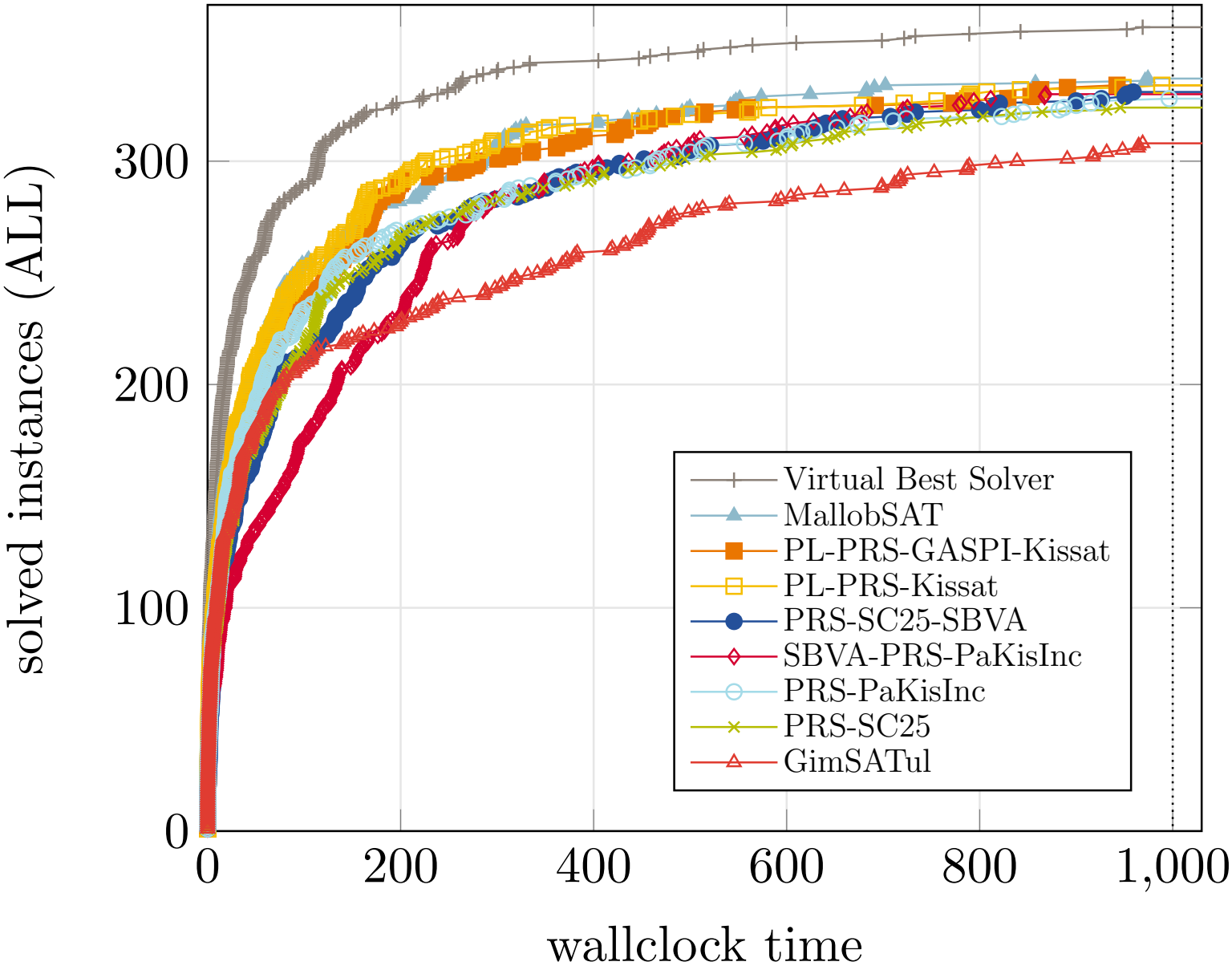
# Parallel Track UNSAT Plot



# Parallel Track

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | MallobSat | Dominik Schreiber, Niccolo Rigi-Luperti, and Armin Biere | 394.56 | 337 |
| 2 | PL-PRS-Kissat | Mazigh Saoudi, Souheib Baarir, Julien Sopena, Thibault Lejemble, and Sabrine Saouli | 406.87 | 334 |
| 3 | PRS-SC25-SBVA | Yuhang Qian, Zhihan Chen, Xindi Zhang, and Shaowei Cai | 457.19 | 331 |

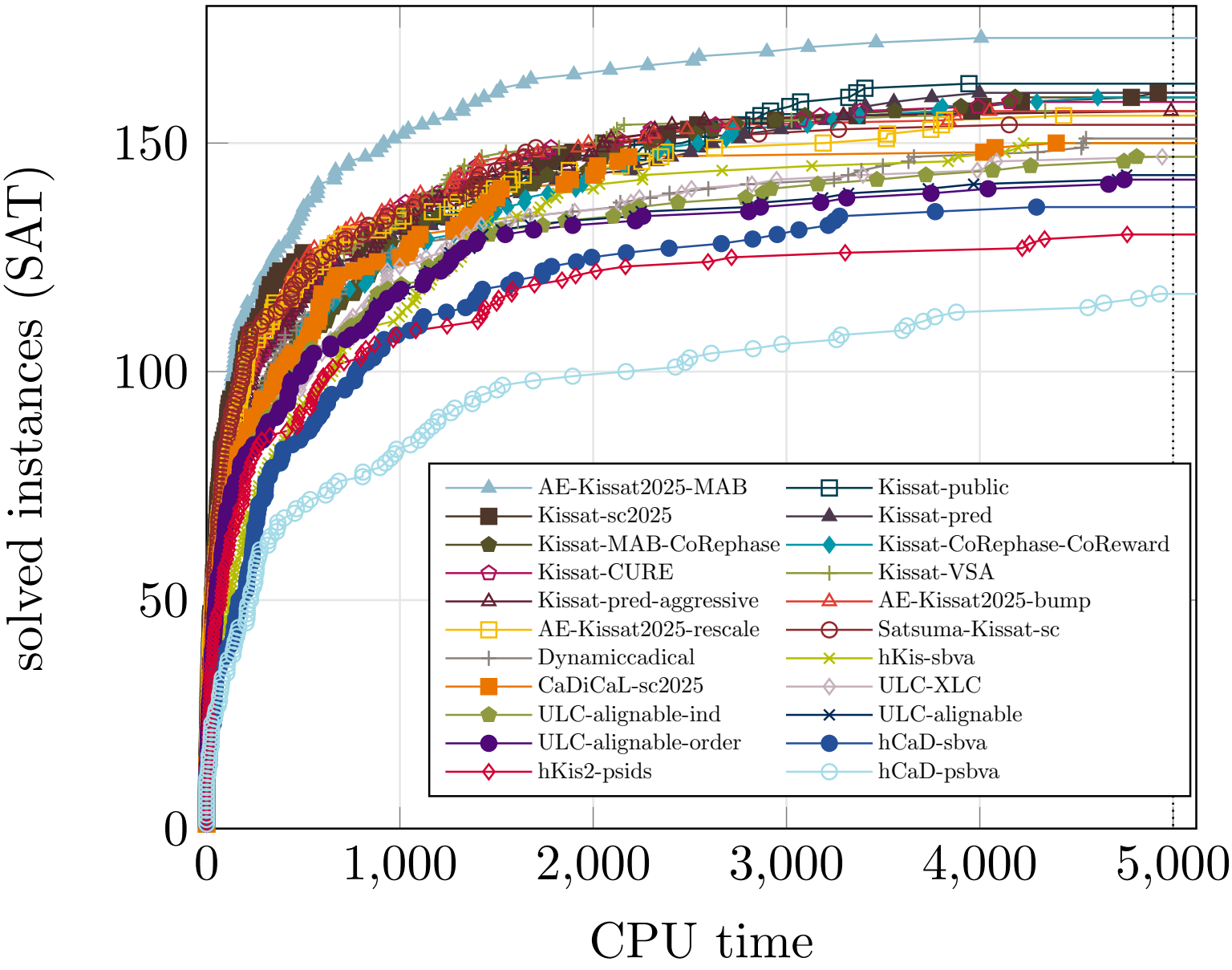
# Parallel Track ALL Plot



# Main (Sequential) Track SAT

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | AE-Kissat-MAB | Hang Ding, Mao Luo, Chu-Min Li, Shunwei Li, Runyao Chen, Caiquan Xiong, and Xinyun Wu | 715.921 | 173 |
| 2 | Kissat-public | Armin Biere, Tobias Faller, Mathias Fleury, Nils Froleyks, and Florian Pollitt | 1397.09 | 163 |
| 3 | Kissat-CURE | Yalun Cai, Mengxia Tao, Kezhi Li, Jiaying Zhu, Zhengyuan Shi, and Qiang Xu | 1525.91 | 159 |

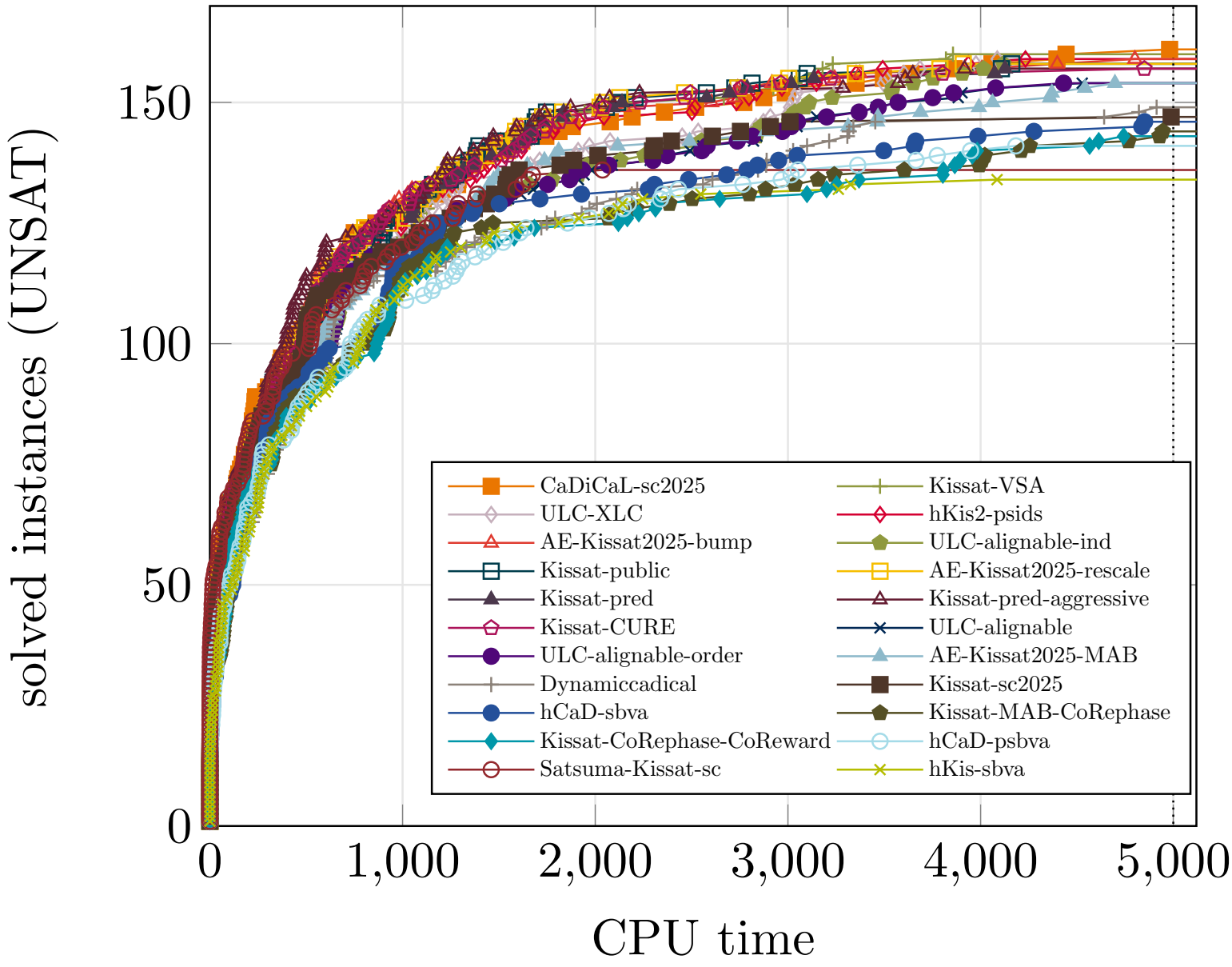
# Main (Sequential) Track SAT Plot



# Main Sequential Track UNSAT

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | CaDiCaL-SC2025 | Armin Biere, Tobias Faller, Mathias Fleury, Nils Froleyks, and Florian Pollitt | 2327.00 | 161 |
| 2 | Kissat-VSA | Shuolin Li and Jialu Zhang | 2335.54 | 160 |
| 3 | AE-Kissat-bump | Hang Ding, Mao Luo, Chu-Min Li, Shunwei Li, Runyao Chen, Caiquan Xiong, Xinyun Wu | 2358.10 | 159 |

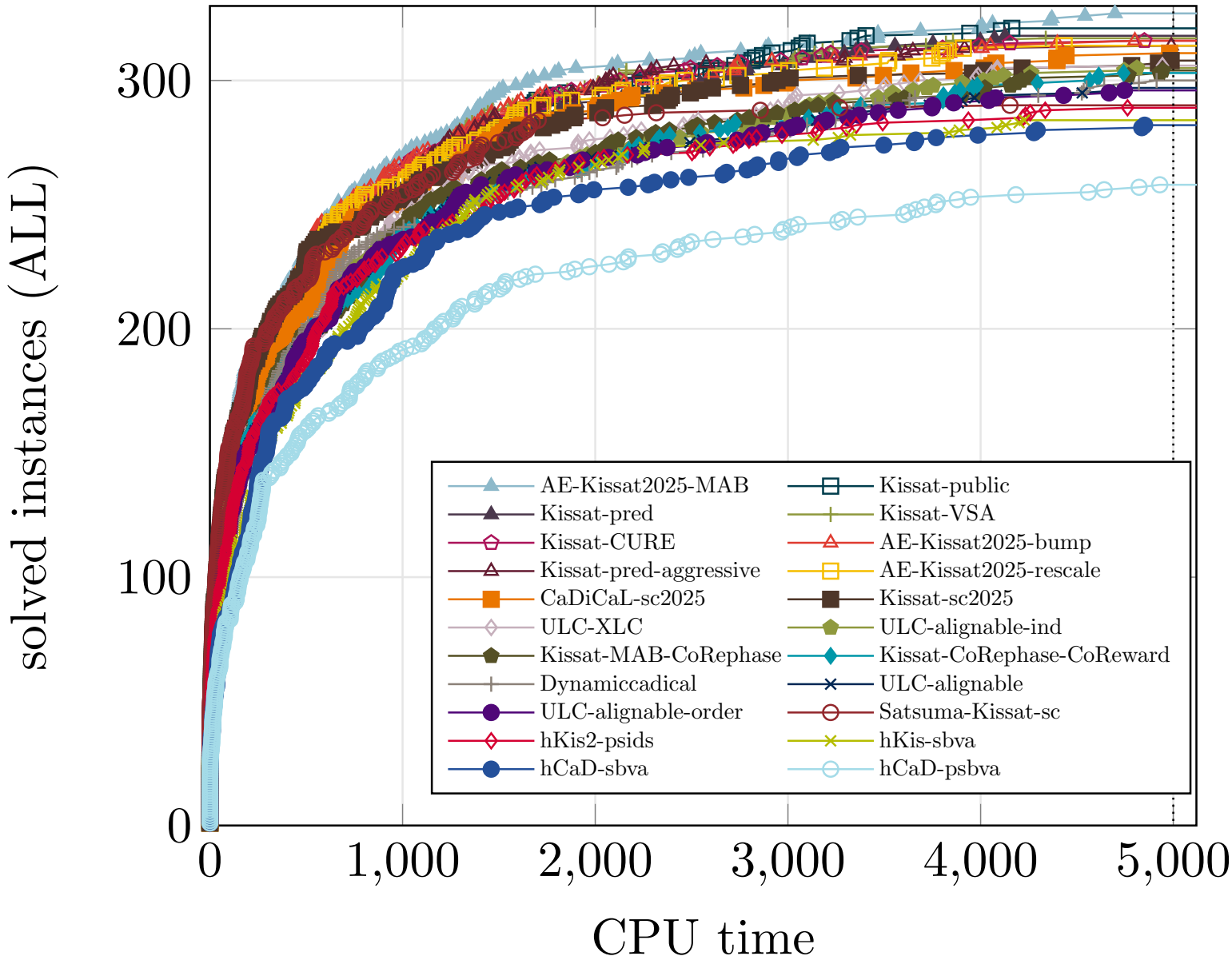
# Main (Sequential) Track UNSAT Plot



# Main Sequential Track

| | Solver | Authors | PAR-2 | Solved |
|---|---|---|---|---|
| 1 | AE-Kissat-MAB | Hang Ding, Mao Luo, Chu-Min Li, Shunwei Li, Runyao Chen, Caiquan Xiong, Xinyun Wu | 2264.73 | 327 |
| 2 | Kissat-public | Armin Biere, Tobias Faller, Mathias Fleury, Nils Froleyks, Florian Pollitt | 2423.38 | 321 |
| 3 | Kissat-VSA | Shuolin Li and Jialu Zhang | 2478.45 | 317 |

# Main (Sequential) Track ALL Plot



# Memory Issues in Main Track

On StarExec, developers were used to 128 GB of memory

The new LMU infrastructure has only 30 GB (as announced)

- Measuring the memory usage is difficult
- Usually, an overapproximation is used

However, several runs crashed at 20 GB allocation

- Parsing issues on some huge instances
- Proof checking issues on some big proofs

# Memory Issues in Main Track

On StarExec, developers were used to 128 GB of memory

The new LMU infrastructure has only 30 GB (as announced)

- Measuring the memory usage is difficult
- Usually, an overapproximation is used

However, several runs crashed at 20 GB allocation

- Parsing issues on some huge instances
- Proof checking issues on some big proofs

What to do next year?

- Limit size of benchmarks?
- More memory?

# Standardize Output

The output of sequential solvers follows the conventions, but this is not the case for parallel / cloud solvers

Exit codes:

- 10: Satisfiable
- 20: Unsatisfiable
- Anything else: error
- Let's establish additional ones: e.g. MEMOUT, TIMEOUT

Result line, starting with `s`, only three options:

- `s SATISFIALBE`: Satisfiable
- `s UNSATISFIALBE`: Unsatisfiable
- `s UNKNOWN` : error

# AWS Infrastucture

AWS values research in parallel and cloud solvers

AWS plans to support parallel/cloud SAT+SMT competitions

- Statement of intent; nothing guaranteed

Existing infrastructure proved difficult for both solver authors and Amazon staff

- (authors) Multiple Docker containers
- (authors) AWS console interactions for configuration
- (authors) Leftover infrastructure could use credits
- (AWS) Debugging insufficiently tested solvers
- (AWS) Inconsistencies in public and internal infrastructure

# New Infrastructure Addresses Previous Issues

- Same code used for development/testing and competition
- AWS configuration, setup, teardown, and job submission managed by a single script
- Author burden is vastly reduced
  1. Provide a Dockerfile that builds the solver image
  2. Single LoC to specify solver entrypoint and options
- Single docker image for distributed leader and workers
- Beta version used for this year's parallel SAT contest

# AWS Pre-Submission Testing and Analysis

- In previous years, AWS effort for solver debug was not sustainable
- Test suite will be provided to solver authors in advance
- Organizers will use the same tests to confirm readiness
- Failing solvers will be disqualified
- Analysis and visualization scripts will be public
- Anyone can simulate the contest (with sufficient budget)

# SAT Comp 2026 AWS Timetable

- Preliminary code and instructions posted in October
- We request community testing
- Feedback to AWS by end of January
- Updates in February and March
- Final[1] code and instructions posted in March

[1] Subsequent changes may be needed; we will strive to minimize the impact on solver authors

# Two More Things...

Solvers using the DIMACS input format since early 2000s.

- Formulas, solutions, proofs, cores, etc are hard to read

Should solvers and related tools support string inputs?

# Two More Things...

Solvers using the DIMACS input format since early 2000s.

- Formulas, solutions, proofs, cores, etc are hard to read

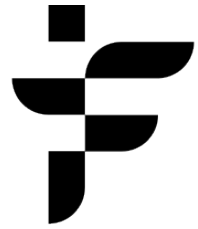
Should solvers and related tools support string inputs?

Proof logging & checking matured over the last decade

- Checking time is roughly similar to solving time
- However, LRAT checking is much faster

Should we have a track that combines solver and checker time?

# Thank you!

FILUTA


Robert Jones (AWS)
Benjamin Jones (AWS)
Philipp Wendler (LMU)