# Cellular Automata Performance Benchmark Report

Auto-generated Analysis

January 28, 2026

## 1 Test Configuration

All benchmarks were executed with the following configuration:

- **Grid Size:** 3000 × 3000 cells (9,000,000 total)
- **Timesteps:** 100
- **Traits:** 1
  - Trait 0: (any) → allowed rules: conway, conway optimized

**Note:** The versions correspond to Git commits but have been modified for benchmarking purposes. The modifications (diffs) are available in the `benchmark_diffs/` folder, with a summary in `benchmark_diffs/SUMMARY.md`.

### 1.1 Configuration Validation

All versions conform to the expected test configuration.

## 2 Library Performance (`trait_ac`)

This section shows the performance of the core simulation library without graphical interface overhead.

| Version | Date | Time (s) | Steps/s | MCells/s | vs Base | vs Prev |
|---|---|---|---|---|---|---|
| v0 | 2025-12-19 | 171.700 | 0.58 | 5.24 | — | — |
| v1 | 2025-12-22 | 26.465 | 3.78 | 34.01 | 6.49× | 6.49× |
| v2 | 2025-12-23 | 26.466 | 3.78 | 34.01 | 6.49× | 1.00× |
| v3 | 2025-12-24 | 12.287 | 8.14 | 73.25 | 13.97× | 2.15× |
| v4 | 2025-12-25 | 1.855 | 53.90 | 485.08 | 92.54× | 6.62× |
| v5 | 2025-12-26 | 1.639 | 61.03 | 549.25 | 104.79× | 1.13× |
| v6 | 2026-01-07 | 0.901 | 111.04 | 999.32 | 190.65× | 1.82× |
| v7 | 2026-01-27 | 0.674 | 148.36 | 1335.21 | 254.73× | 1.34× |

Table 1: Library performance across versions

**Summary:** Final version achieves a **254.7× speedup** compared to baseline. Time reduced from 171.700s to 0.674s.

## 3 Graphical Interface Performance (`trait_ac_ui`)

This section shows the performance including the graphical rendering overhead.

| Version | Date | Time (s) | Steps/s | MCells/s | vs Base | vs Prev |
|---|---|---|---|---|---|---|
| v0 | 2025-12-19 | 540.806 | 0.18 | 1.66 | — | — |
| v1 | 2025-12-22 | 320.368 | 0.31 | 2.81 | 1.69× | 1.69× |
| v2 | 2025-12-23 | 32.625 | 3.07 | 27.59 | 16.58× | 9.82× |
| v3 | 2025-12-24 | 21.195 | 4.72 | 42.46 | 25.52× | 1.54× |
| v4 | 2025-12-25 | 7.371 | 13.57 | 122.10 | 73.37× | 2.88× |
| v5 | 2025-12-26 | 4.277 | 23.38 | 210.41 | 126.44× | 1.72× |
| v6 | 2026-01-07 | 2.262 | 44.22 | 397.94 | 239.12× | 1.89× |
| v7 | 2026-01-27 | 0.913 | 109.59 | 986.28 | 592.65× | 2.48× |

Table 2: GUI performance across versions

**Summary:** Final version achieves a **592.7× speedup** compared to baseline. Time reduced from 540.806s to 0.913s.

## 4 UI vs Library Time Breakdown

This analysis separates the pure UI rendering overhead from the simulation computation time.

| Version | Date | UI Total (s) | Lib (s) | Pure UI (s) | Lib % | UI % | Pure UI Speedup |
|---|---|---|---|---|---|---|---|
| v0 | 2025-12-19 | 540.806 | 171.700 | 369.106 | 31.7% | 68.3% | — |
| v1 | 2025-12-22 | 320.368 | 26.465 | 293.903 | 8.3% | 91.7% | 1.26× |
| v2 | 2025-12-23 | 32.625 | 26.466 | 6.159 | 81.1% | 18.9% | 59.93× |
| v3 | 2025-12-24 | 21.195 | 12.287 | 8.908 | 58.0% | 42.0% | 41.43× |
| v4 | 2025-12-25 | 7.371 | 1.855 | 5.516 | 25.2% | 74.8% | 66.92× |
| v5 | 2025-12-26 | 4.277 | 1.639 | 2.639 | 38.3% | 61.7% | 139.88× |
| v6 | 2026-01-07 | 2.262 | 0.901 | 1.361 | 39.8% | 60.2% | 271.19× |
| v7 | 2026-01-27 | 0.913 | 0.674 | 0.238 | 73.9% | 26.1% | 1547.82× |

Table 3: UI overhead breakdown by version

**Pure UI Overhead Improvement:** 1547.8× speedup (from 369.106s to 0.238s)

## 5 Version History

| Version | Commit | Date | Message |
|---|---|---|---|
| v0 | `d3c5067` | 2025-12-19 | update readme.md |
| v1 | `d76218f` | 2025-12-22 | performance boost |
| v2 | `6002d09` | 2025-12-23 | performance boost + refactor |
| v3 | `4e7ee8c` | 2025-12-24 | perf boost (less allocation + new approach) |
| v4 | `c1d129c` | 2025-12-25 | refactor + better perf |
| v5 | `de2a57f` | 2025-12-26 | gpu rendering |
| v6 | `d5b73d0` | 2026-01-07 | refactor, cleanup, fixed small bugs, added conf... |
| v7 | `95c8875` | 2026-01-27 | removed bitvec, updated conway_optimized, upda... |

Table 4: Git commit history for each version

# 6 Overall Summary

| Component | Baseline Time | Final Speedup |
|---|---|---|
| Library (`trait_ac`) | 171.700s | 254.7× |
| GUI (`trait_ac_ui`) | 540.806s | 592.7× |
| Pure UI Overhead | 369.106s | 1547.8× |

The optimization effort resulted in a **255× speedup** for the core library, going from 5.24 million cells/second to 1335.21 million cells/second (**1.34 billion cells/second**).