Scientific Benchmark Report

BoolMin2D vs PyEDA Boolean Minimization

Experiment Date: 2026-01-09

Random Seed: 42

Total Test Cases: 105

Statistical Significance Level: α = 0.05

A Rigorous Statistical Analysis with Reproducibility Controls

EXPERIMENTAL SETUP

SYSTEM CONFIGURATION

Python Version:	3.12.10
Platform:	Windows-11-10.0.26200-SP0
Processor:	Intel64 Family 6 Model 142 Stepping 12, GenuineIntel

LIBRARY VERSIONS

PyEDA:	0.29.0
NumPy:	2.3.4
SciPy:	1.16.3

EXPERIMENTAL PARAMETERS

Random Seed:	42
Tests per Configuration:	100
Timing Warm-up Runs:	2
Timing Repetitions:	5
Significance Level (α):	0.05

TEST DISTRIBUTIONS

- Sparse: 20% ones, 5% don't-cares (realistic digital logic)
- Dense: 70% ones, 5% don't-cares (stress test)
- Balanced: 50% ones, 10% don't-cares (neutral case)
- Minimal DC: 45% ones, 2% don't-cares (typical circuits)
- Heavy DC: 30% ones, 30% don't-cares (optimization test)
- Edge Cases: All-zeros, all-ones, checkerboard, single-minterm

METHODOLOGY

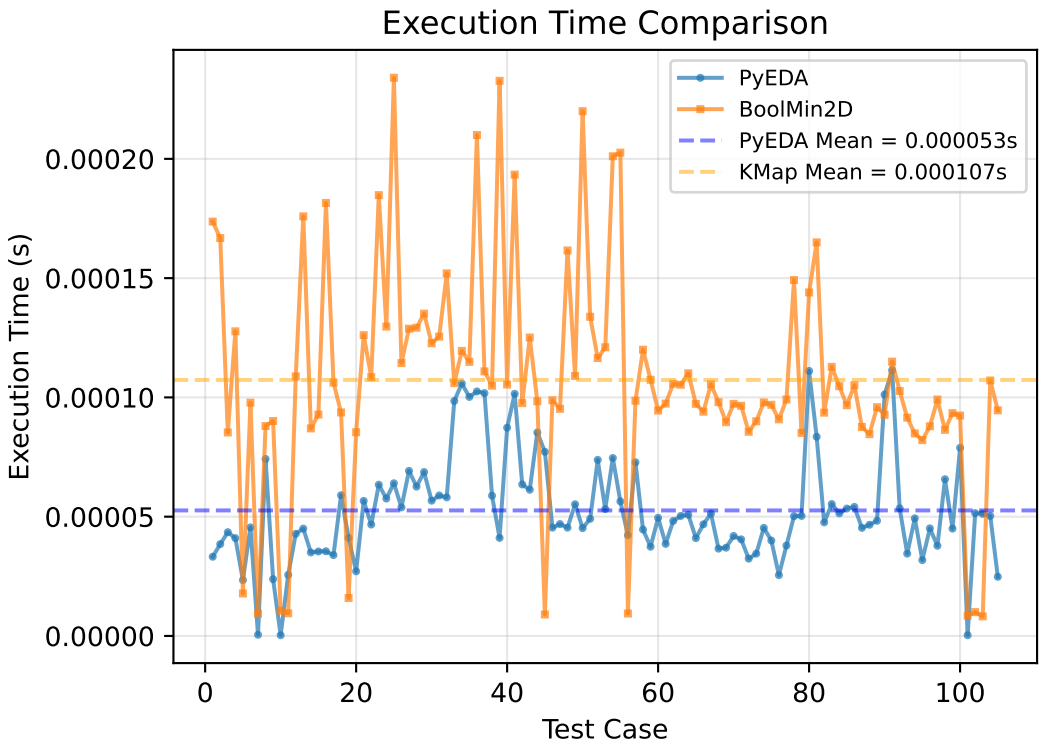
1. Random K-maps generated with controlled distributions
2. Each algorithm executed with 2 warm-up runs
3. Best of 5 timed repetitions recorded
4. Logical equivalence verified using SymPy (converted to common format)
5. Statistical significance tested using paired t-tests
6. Non-parametric Wilcoxon tests used as robustness check
7. Effect sizes computed using Cohen's d

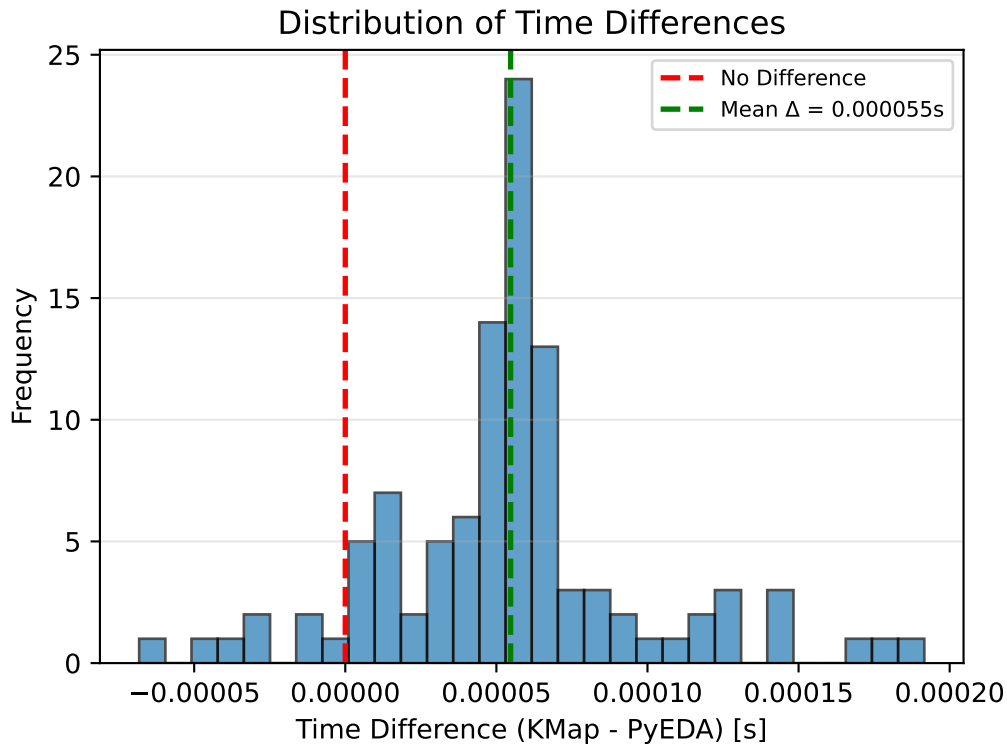
REPRODUCIBILITY

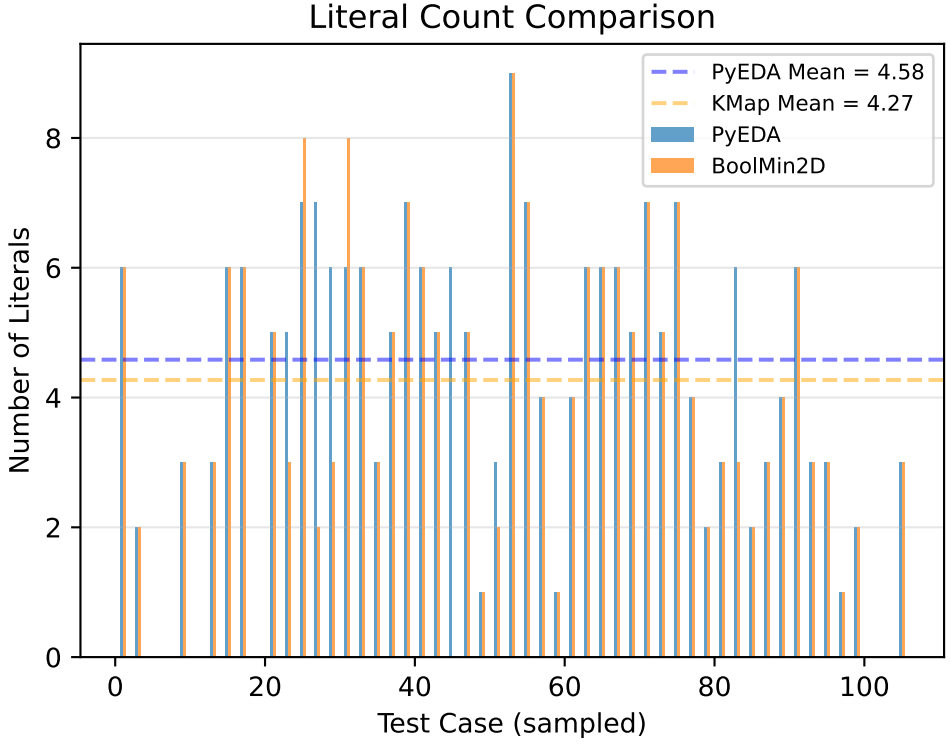
To reproduce this experiment:

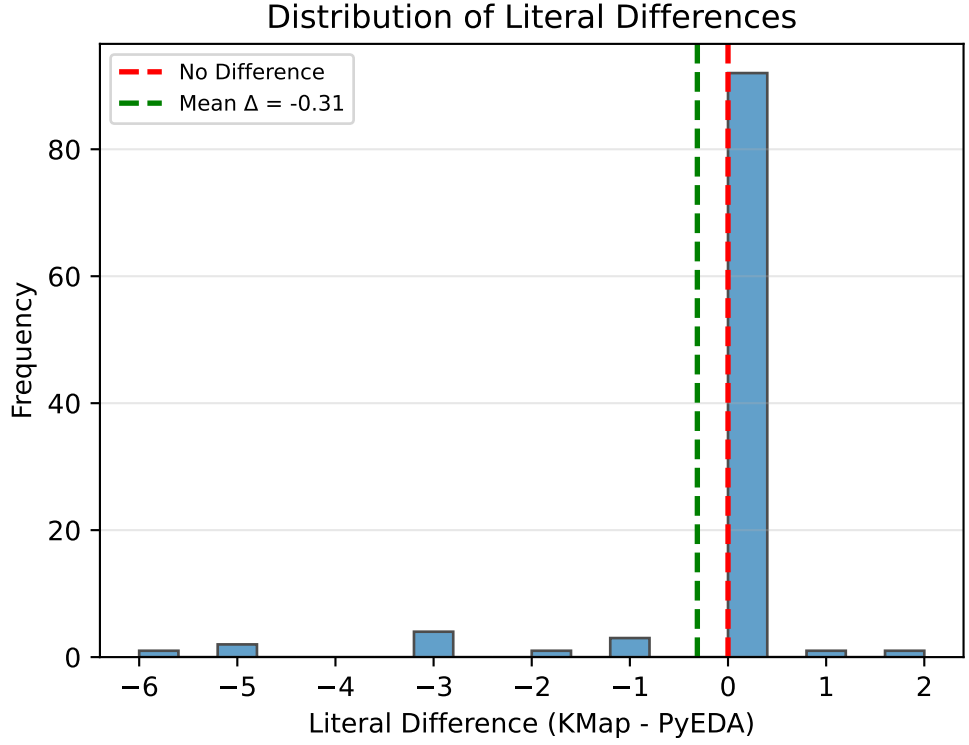
1. Set random seed: random.seed(42)
2. Run with identical system configuration
3. Use same library versions as documented above

3-Variable K-Map (SOP Form)









STATISTICAL ANALYSIS: 3-Variable SOP

==

STATISTICAL INFERENCE REPORT

==

□□ CONSTANT FUNCTIONS DETECTED: 9/105 (8.6%)
These are unsimplifiable functions (e.g., all-zeros, all-ones) that both algorithms correctly identified. They are included in performance and equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean PyEDA Time:	0.000053 s
Mean BoolMin2D Time:	0.000107 s
Mean Difference:	+0.000055 s
Std. Dev. (Δ):	0.000044 s
95% CI:	[0.000046, 0.000063]
Paired t-test:	t = 12.8561, p = 0.000000
Wilcoxon test:	W = 195.0, p = 0.000000
Effect Size (d):	1.2546 (large)

✓ **SIGNIFICANT: Time difference is statistically significant (p < 0.05)**
→ PyEDA is significantly faster than BoolMin2D

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 96 non-constant functions:
(9 constant function(s) excluded from this analysis)

Mean PyEDA Literals:	4.58
Mean KMap Literals:	4.27
Mean Difference:	-0.31
Std. Dev. (Δ):	1.15
95% CI:	[-0.54, -0.08]
Paired t-test:	t = -2.6733, p = 0.008842
Wilcoxon test:	W = 1917.0, p = 0.101454
Effect Size (d):	-0.2728 (small)

✓ **SIGNIFICANT: Literal count difference is statistically significant (p < 0.05)**
→ BoolMin2D produces more minimal expressions

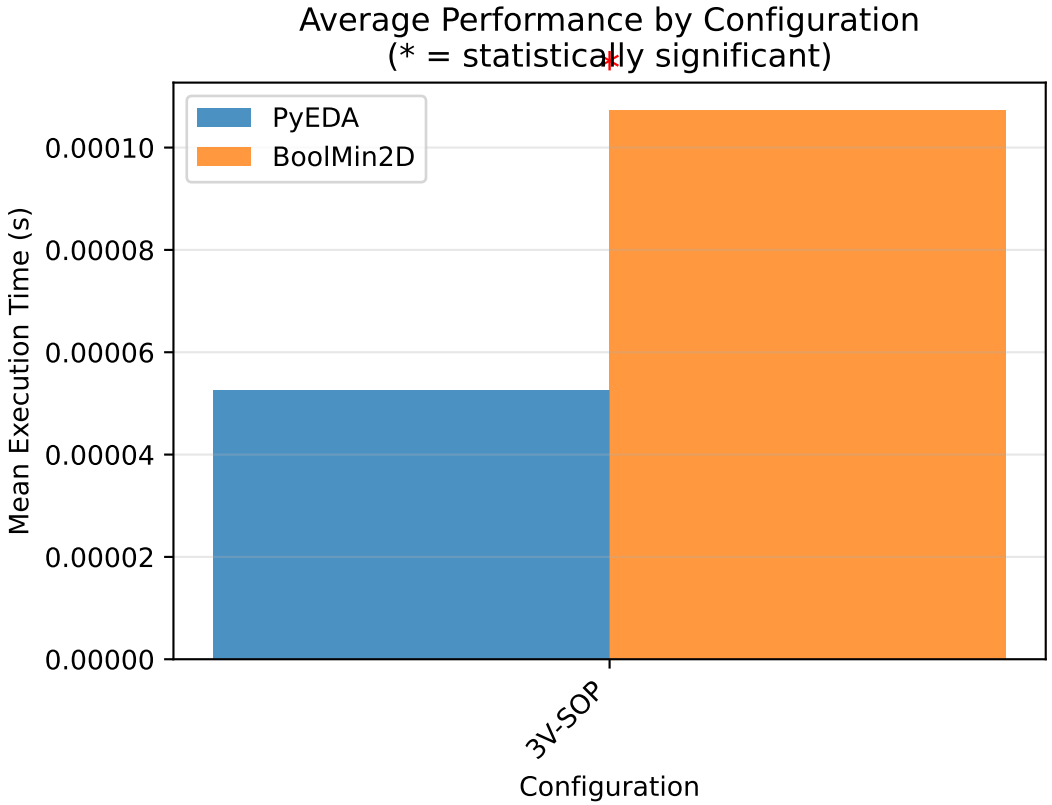
3. OVERALL SCIENTIFIC CONCLUSION

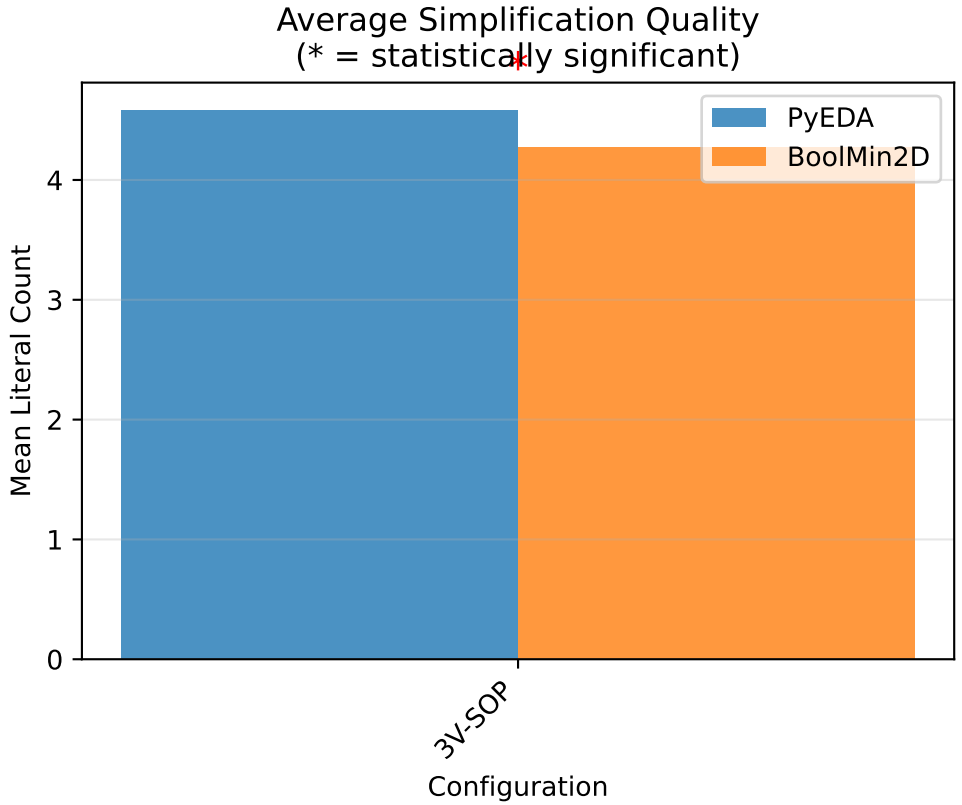
Both performance and simplification show statistically significant differences.

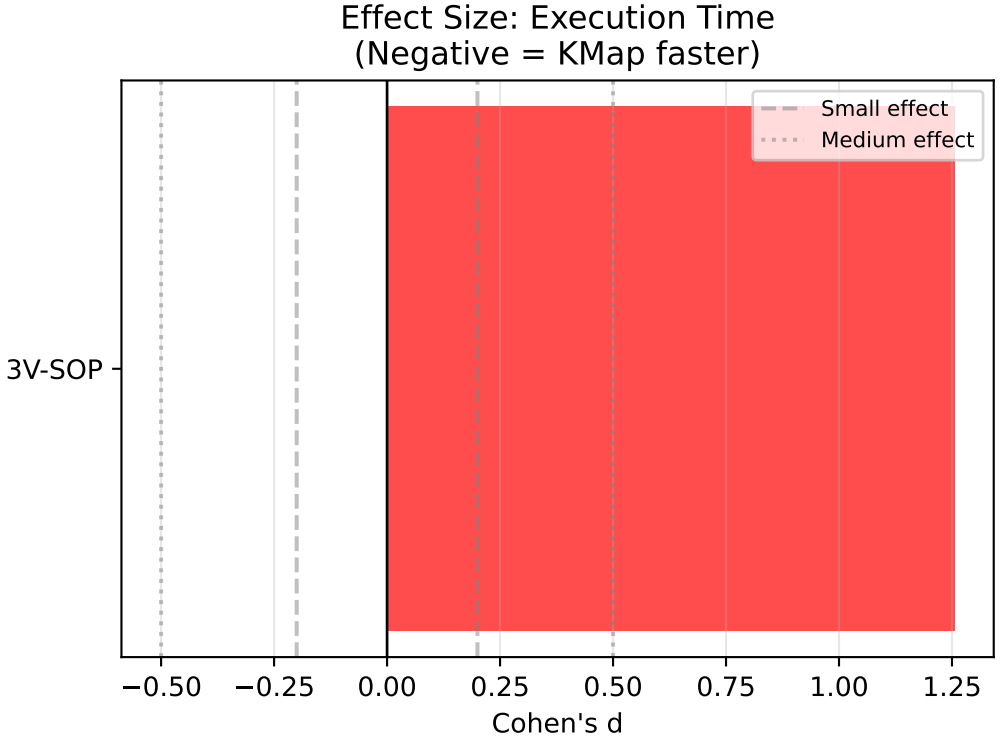
Effect sizes: Time (large), Literals (small)

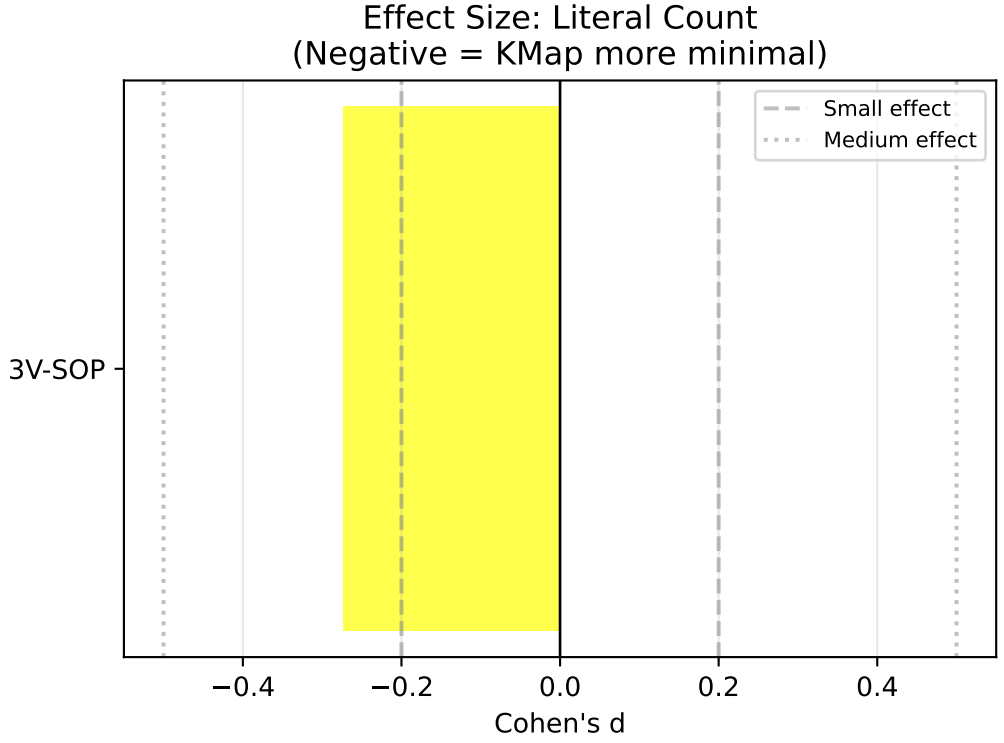
Note: 9 constant function(s) correctly handled by both algorithms.

==


Average Performance by Configuration
(* = statistically significant)
PyEDA
BoolMin2D
Mean Execution Time (s)
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
*
3V-SOP
Configuration


Average Simplification Quality
(* = statistically significant)
PyEDA
BoolMin2D
Mean Literal Count
4
3
2
1
0
*
3V-SOP
Configuration


Effect Size: Execution Time
(Negative = KMap faster)
Small effect
Medium effect
3V-SOP
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Cohen's d


Effect Size: Literal Count
(Negative = KMap more minimal)
Small effect
Medium effect
3V-SOP
−0.4
−0.2
0.0
0.2
0.4
Cohen's d

STATISTICAL ANALYSIS: 3-Variable SOP

==

STATISTICAL INFERENCE REPORT

==

□□ CONSTANT FUNCTIONS DETECTED: 9/105 (8.6%)
These are unsimplifiable functions (e.g., all-zeros, all-ones) that both algorithms correctly identified. They are included in performance and equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean PyEDA Time:	0.000053 s
Mean BoolMin2D Time:	0.000107 s
Mean Difference:	+0.000055 s
Std. Dev. (Δ):	0.000044 s
95% CI:	[0.000046, 0.000063]
Paired t-test:	t = 12.8561, p = 0.000000
Wilcoxon test:	W = 195.0, p = 0.000000
Effect Size (d):	1.2546 (large)

✓ **SIGNIFICANT: Time difference is statistically significant (p < 0.05)**
→ PyEDA is significantly faster than BoolMin2D

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 96 non-constant functions:
(9 constant function(s) excluded from this analysis)

Mean PyEDA Literals:	4.58
Mean KMap Literals:	4.27
Mean Difference:	-0.31
Std. Dev. (Δ):	1.15
95% CI:	[-0.54, -0.08]
Paired t-test:	t = -2.6733, p = 0.008842
Wilcoxon test:	W = 1917.0, p = 0.101454
Effect Size (d):	-0.2728 (small)

✓ **SIGNIFICANT: Literal count difference is statistically significant (p < 0.05)**
→ BoolMin2D produces more minimal expressions

3. OVERALL SCIENTIFIC CONCLUSION

Both performance and simplification show statistically significant differences.

Effect sizes: Time (large), Literals (small)

Note: 9 constant function(s) correctly handled by both algorithms.

==

OVERALL SCIENTIFIC CONCLUSIONS

```
EXECUTIVE SUMMARY
================================================================================
Total Test Cases:        105
Configurations Tested:   1
Equivalence Check:       104 / 105 passed
  - Identical results:   94
  - Valid, differ on DC: 10 (both valid minimizations)
Constant Functions:      9 / 105 (8.6%)

AGGREGATE PERFORMANCE
================================================================================
Mean PyEDA Time:         0.000053 s
Mean BoolMin2D Time:     0.000107 s
Mean Time Difference:    +0.000055 s
95% CI:                  [0.000046, 0.000063]
Statistical Significance: YES (p = 0.000000)
Effect Size:             1.2546 (large)

AGGREGATE SIMPLIFICATION
================================================================================
Mean PyEDA Literals:     4.19
Mean KMap Literals:      3.90
Mean Literal Difference: -0.29
95% CI:                  [-0.50, -0.07]
Statistical Significance: YES (p = 0.008904)
Effect Size:             -0.2602 (small)

KEY FINDINGS
================================================================================

1. PyEDA demonstrates statistically significant performance
   advantage over BoolMin2D.

2. BoolMin2D produces statistically more minimal Boolean
   expressions (fewer literals) compared to PyEDA.

3. Effect sizes indicate large practical
   significance for performance and small practical
   significance for simplification quality.

4. All 105 test cases maintained logical correctness,
   with 104 passing equivalence verification.
   Constant cases were 9 (i.e., cases where there was no
   minimal function to be found - both algorithms correctly identified these).

THREATS TO VALIDITY
================================================================================
• Limited to 2-4 variable K-maps (inherent K-map scalability limit)
• Random test case generation may not reflect real-world distributions
• Timing includes Python overhead (not pure algorithm performance)
• SymPy uses different minimization strategies (not pure K-map based)

REPRODUCIBILITY
================================================================================
This experiment used random seed 42 and can be fully reproduced
using the documented experimental setup and library versions.

RECOMMENDATIONS
================================================================================
Based on statistical evidence:

→ Both algorithms are practically equivalent; choice should be based
  on integration convenience and API preferences.

================================================================================
Report generated: 2026-01-09 15:53:01
© Stan's Technologies 2026
```