



Stan's Technologies

Scientific Benchmark Report

BoolMinGeo vs SymPy (5-8 Variables)

3D geometric approach vs symbolic simplification

Experiment Date: 2026-01-07

Random Seed: 42

Total Test Cases: 80

Statistical Significance Level: $\alpha = 0.05$

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

Sympy: 1.14.0
NumPy: 2.3.4
SciPy: 1.16.3

EXPERIMENTAL PARAMETERS

Random Seed: 42
Tests per Distribution: 1
Tests per Configuration: 10
Timing Warm-up Runs: 1
Timing Repetitions: 3
Significance Level (α): 0.05

TEST CONFIGURATIONS

- 5-variable K-maps (32 minterms)
- 6-variable K-maps (64 minterms)
- 7-variable K-maps (128 minterms)
- 8-variable K-maps (256 minterms)

METHODOLOGY

1. Random and pattern-based test cases generated
2. Each algorithm executed with 1 warm-up runs
3. Best of 3 timed repetitions recorded
4. Logical equivalence verified using Sympy
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

TRIVIAL CONSTANT CASES

Constant functions (all-zeros→False, all-ones→True, all-dc) are already maximally simplified. Both algorithms correctly identify these degenerate cases. They are excluded from literal-count statistics but included in performance and equivalence analysis.

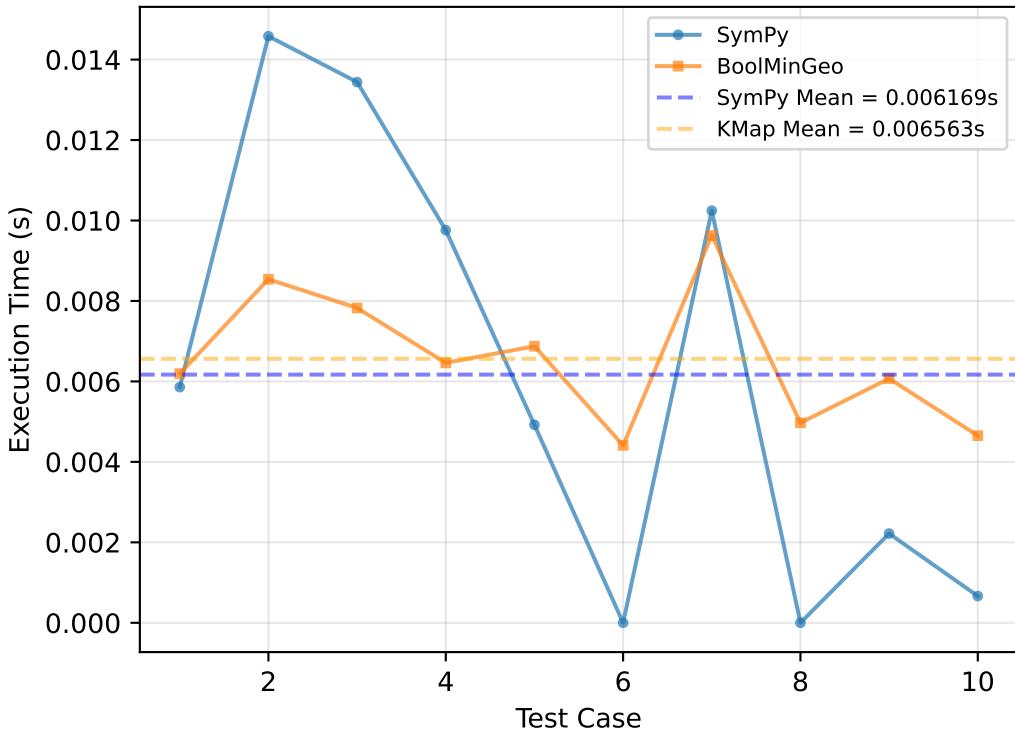
REPRODUCIBILITY

To reproduce this experiment:

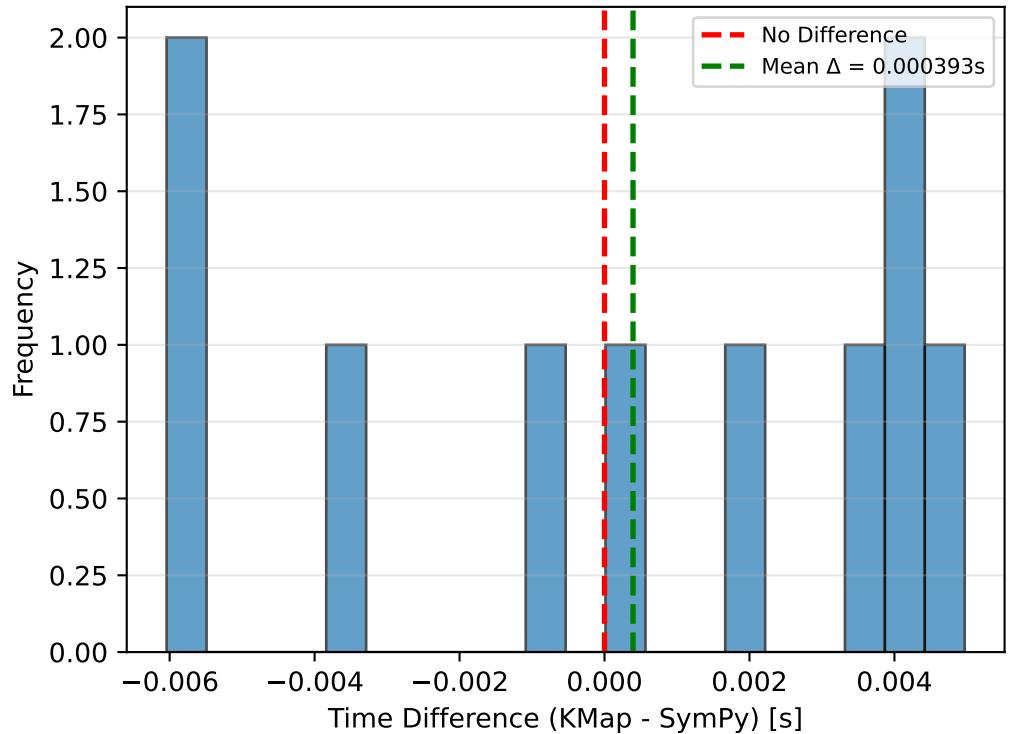
1. Set random seed: `random.seed(42)`
2. Run with identical system configuration
3. Use same library versions as documented above

5-Variable K-Map (SOP Form)

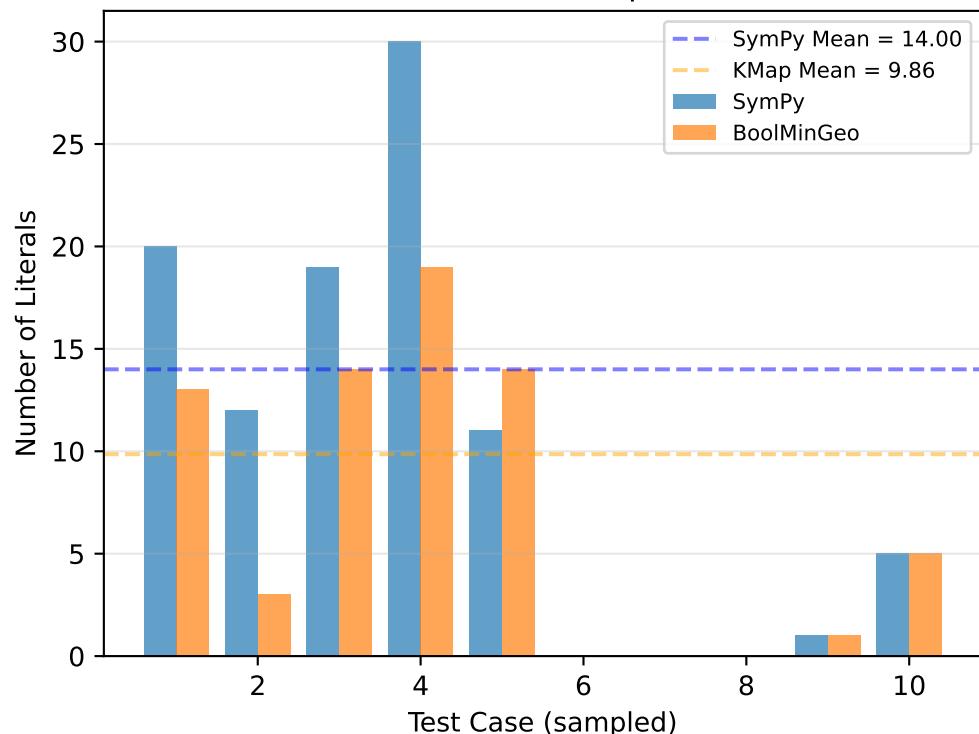
Execution Time Comparison



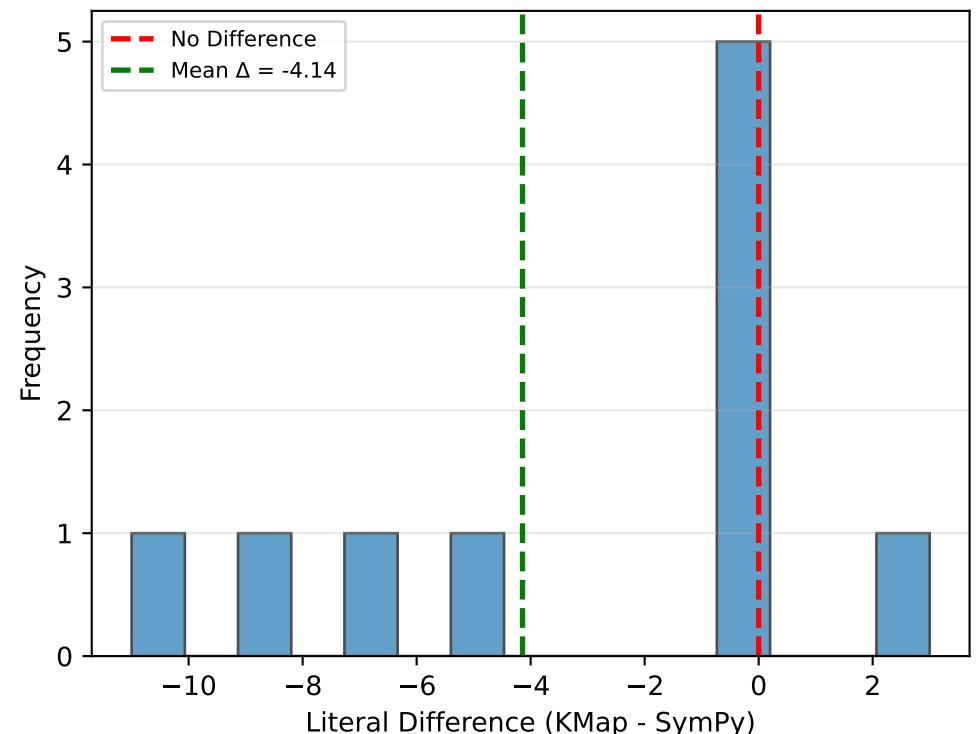
Distribution of Time Differences



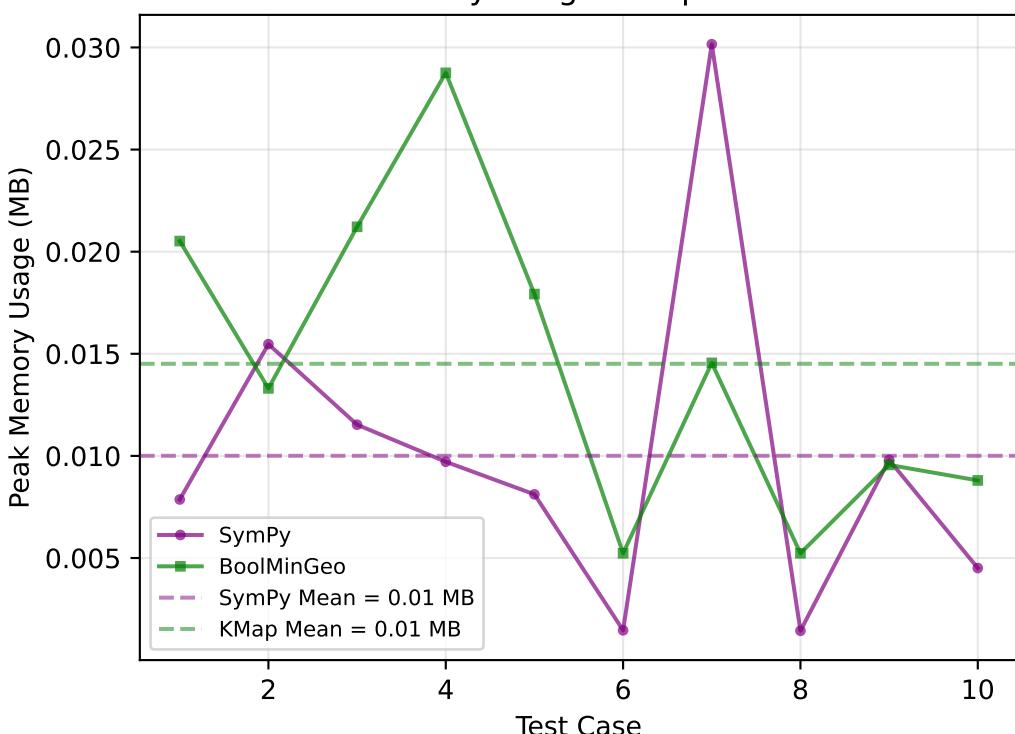
Literal Count Comparison



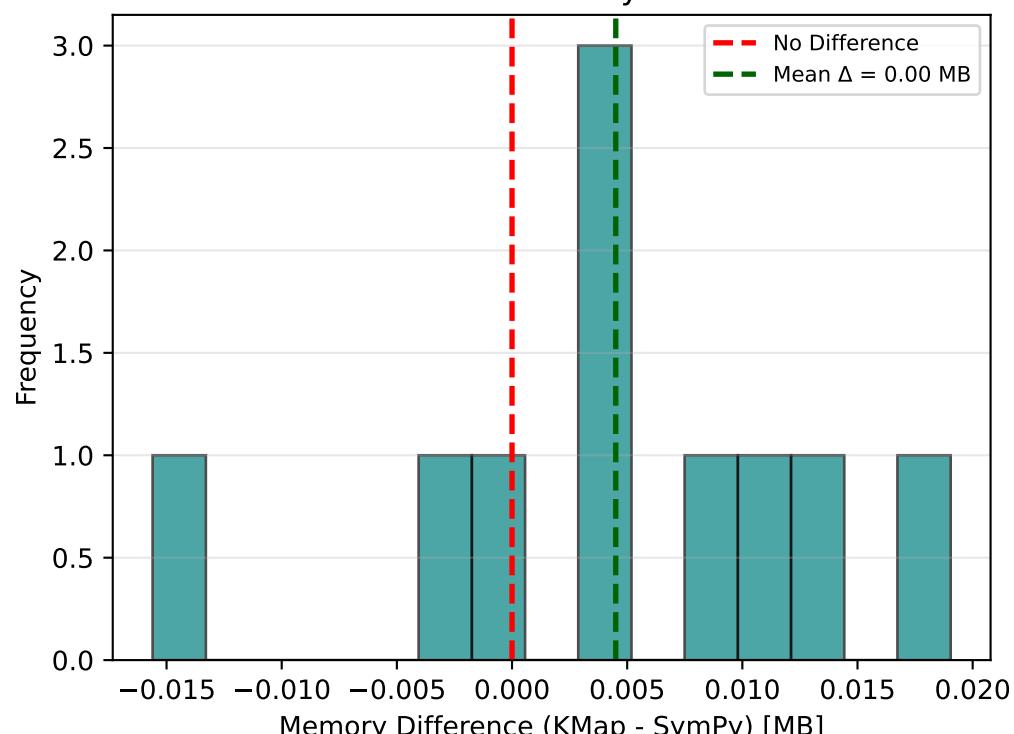
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 5-Variable K-Map (SOP Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)
These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time:	0.006169 s
Mean BoolMinGeo Time:	0.006563 s
Mean Difference:	+0.000393 s
Std. Dev. (Δ):	0.004171 s
95% CI:	[-0.002590, 0.003377]
Paired t-test:	$t = 0.2983, p = 0.772265$
Wilcoxon test:	$W = 25.0, p = 0.845703$
Effect Size (d):	0.0943 (negligible)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:
(3 constant function(s) excluded from this analysis)

Mean SymPy Literals:	14.00
Mean KMap Literals:	9.86
Mean Difference:	-4.14
Std. Dev. (Δ):	5.24
95% CI:	[-8.99, 0.70]
Paired t-test:	$t = -2.0911, p = 0.081476$
Wilcoxon test:	$W = 4.5, p = 0.125000$
Effect Size (d):	-0.7904 (medium)

✗ NOT SIGNIFICANT: No significant difference in simplification ($p \geq 0.05$)
→ Both algorithms achieve comparable minimization

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

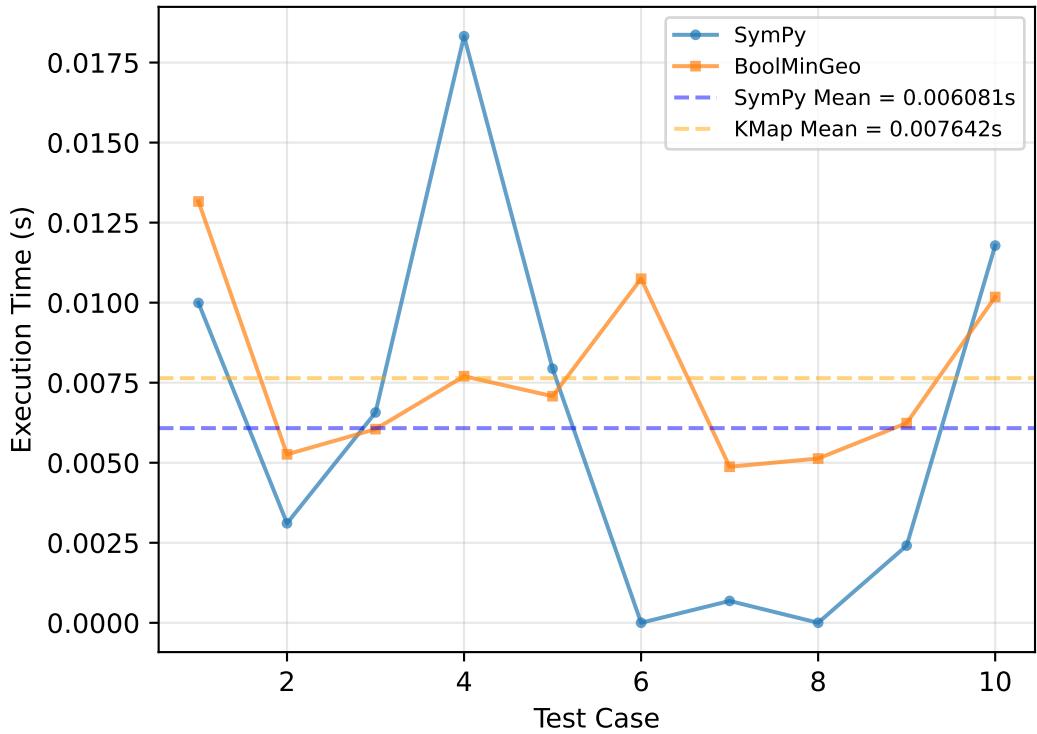
Mean SymPy Memory:	0.01 MB
Mean KMap Memory:	0.01 MB
Mean Difference:	+0.00 MB
Std. Dev. (Δ):	0.01 MB
95% CI:	[-0.00, 0.01]
Paired t-test:	$t = 1.5058, p = 0.166394$
Wilcoxon test:	$W = 12.0, p = 0.130859$
Effect Size (d):	0.4762 (small)

Memory Efficiency: 0.69×
→ SymPy uses 69.0% of BoolMinGeo's memory

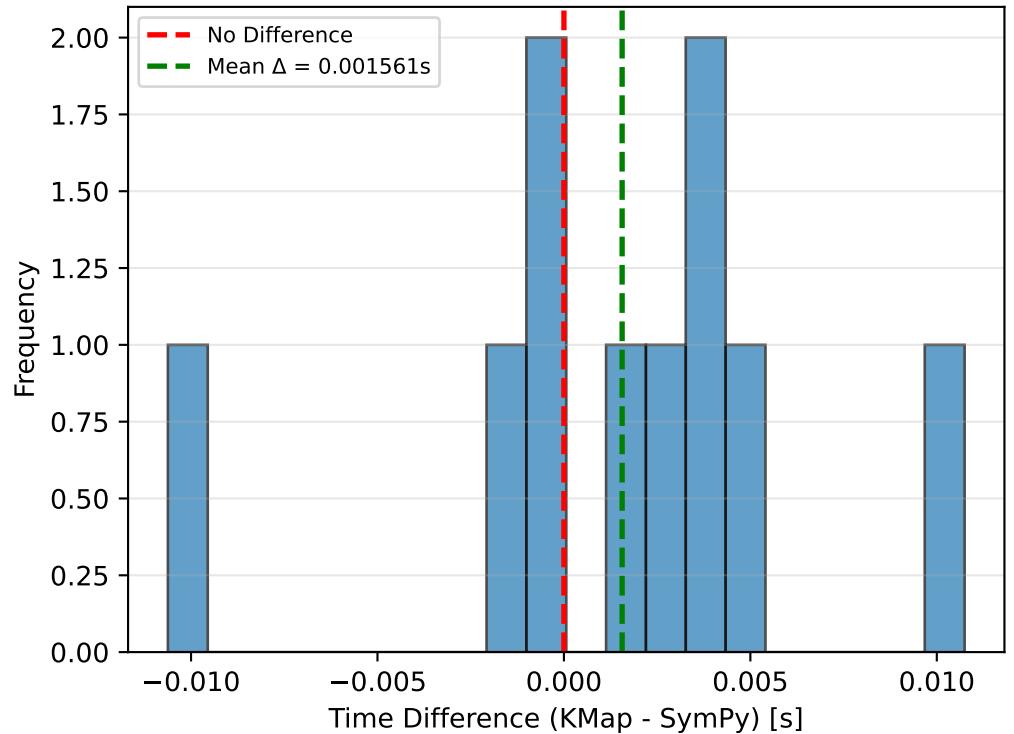
✗ NOT SIGNIFICANT: No significant memory difference ($p \geq 0.05$)
→ Both algorithms have comparable memory usage

5-Variable K-Map (POS Form)

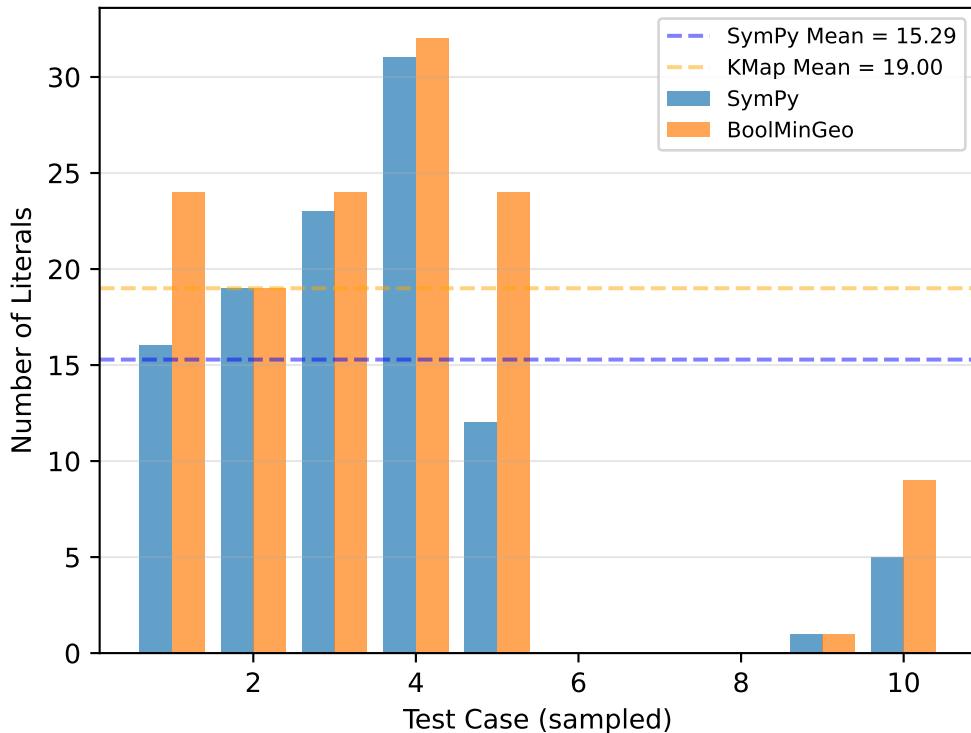
Execution Time Comparison



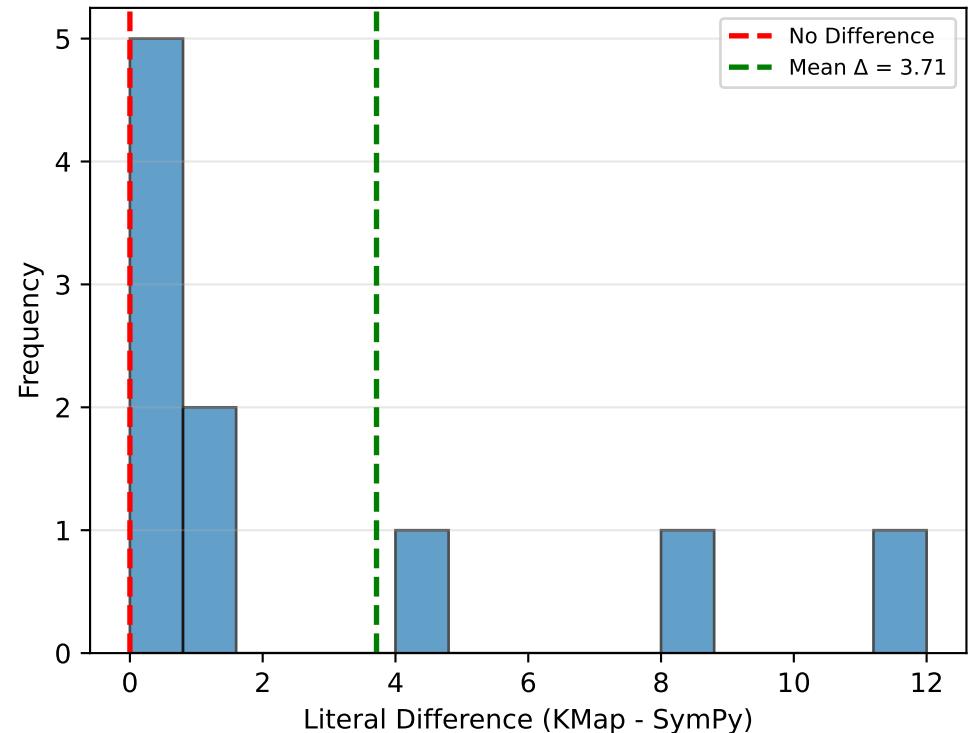
Distribution of Time Differences



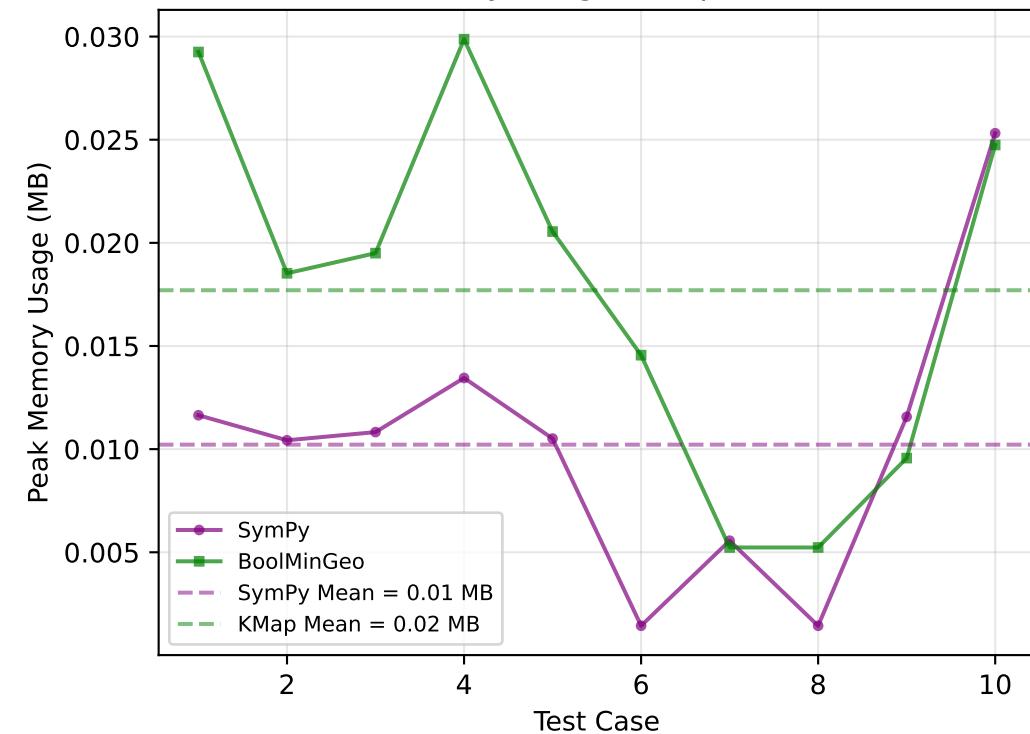
Literal Count Comparison



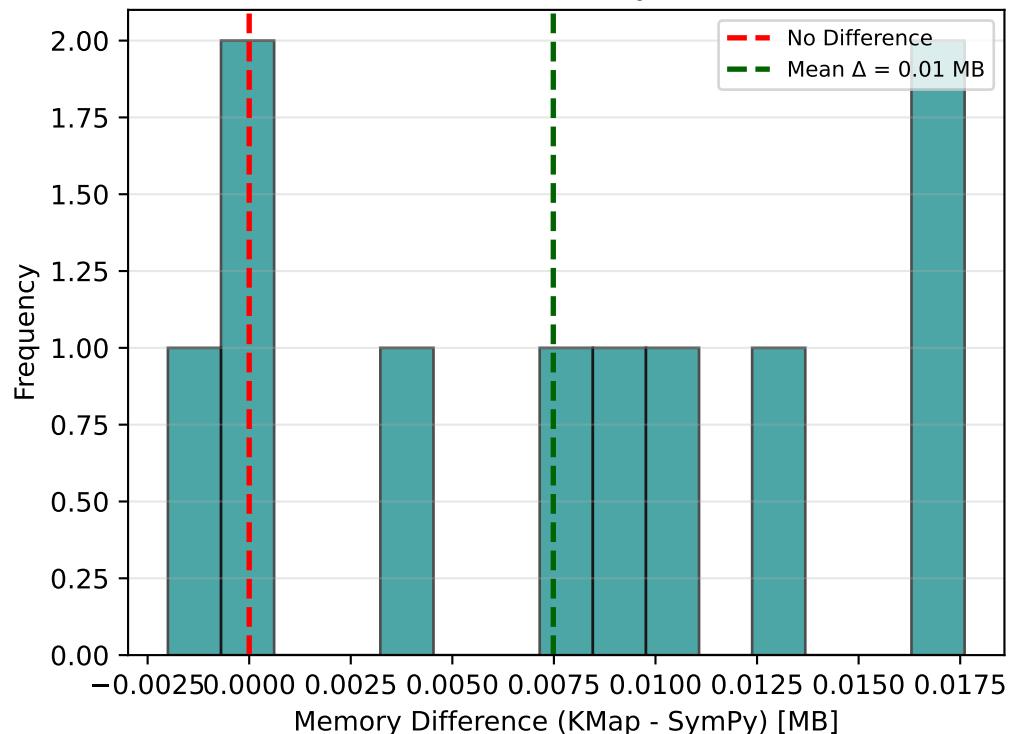
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 5-Variable K-Map (POS Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 0.006081 s

Mean BoolMinGeo Time: 0.007642 s

Mean Difference: +0.001561 s

Std. Dev. (Δ): 0.005585 s

95% CI: [-0.002434, 0.005556]

Paired t-test: $t = 0.8840$, $p = 0.399691$

Wilcoxon test: $W = 15.0$, $p = 0.232422$

Effect Size (d): 0.2795 (small)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 15.29

Mean KMap Literals: 19.00

Mean Difference: +3.71

Std. Dev. (Δ): 4.64

95% CI: [-0.58, 8.01]

Paired t-test: $t = 2.1158$, $p = 0.078734$

Wilcoxon test: $W = 1.5$, $p = 0.062500$

Effect Size (d): 0.7997 (medium)

✗ NOT SIGNIFICANT: No significant difference in simplification ($p \geq 0.05$)
→ Both algorithms achieve comparable minimization

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.01 MB

Mean KMap Memory: 0.02 MB

Mean Difference: +0.01 MB

Std. Dev. (Δ): 0.01 MB

95% CI: [0.00, 0.01]

Paired t-test: $t = 3.3403$, $p = 0.008656$

Wilcoxon test: $W = 6.0$, $p = 0.027344$

Effect Size (d): 1.0563 (large)

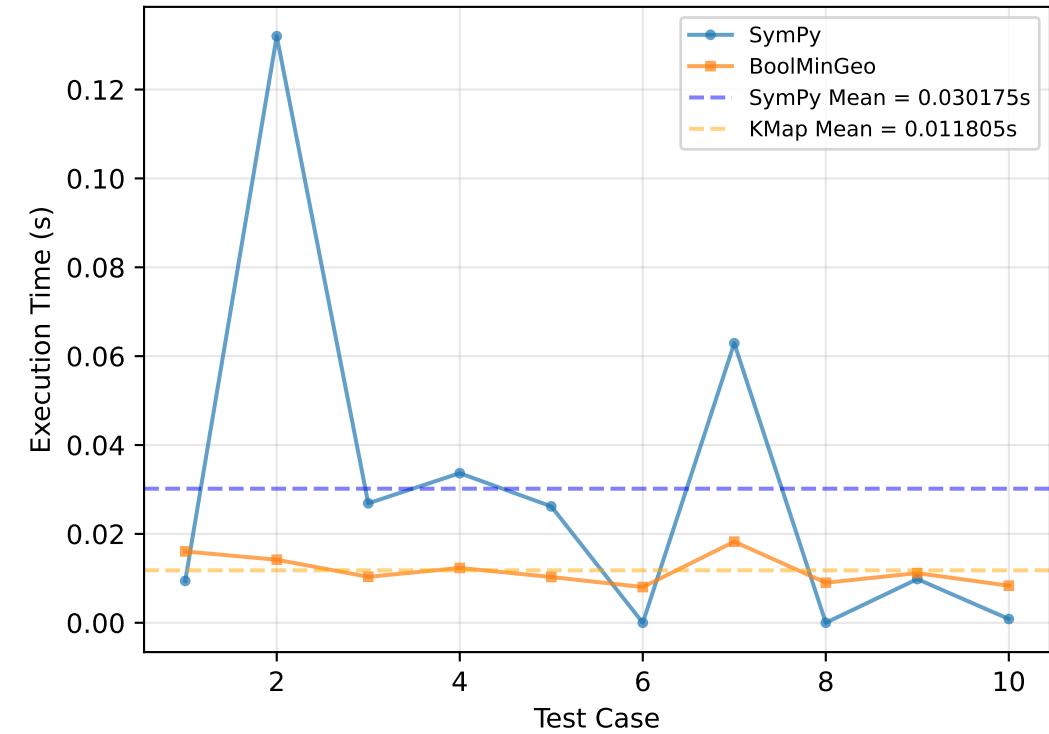
Memory Efficiency: 0.58x

→ SymPy uses 57.7% of BoolMinGeo's memory

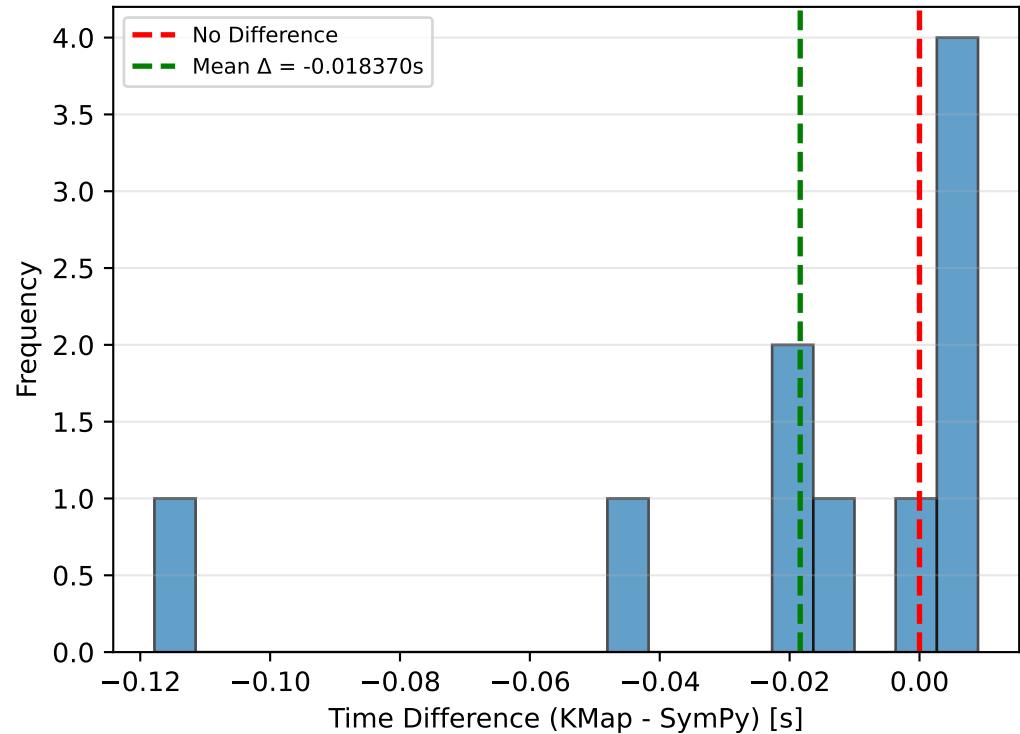
✓ SIGNIFICANT: Memory difference is statistically significant ($p < 0.05$)
→ SymPy uses significantly less memory

6-Variable K-Map (SOP Form)

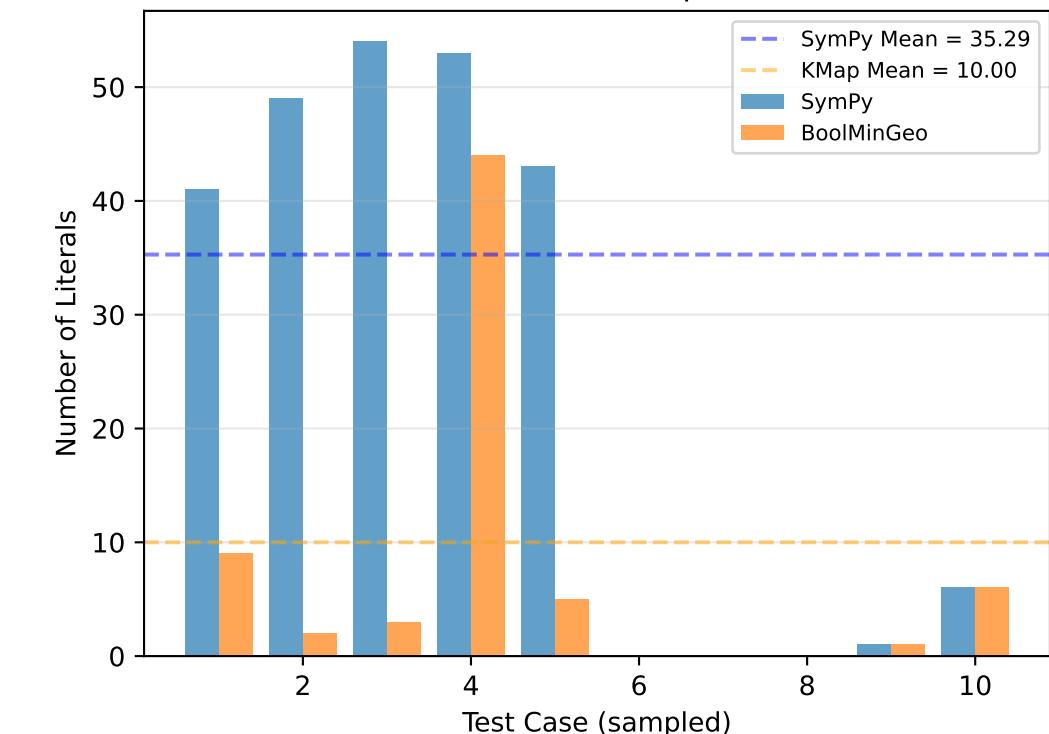
Execution Time Comparison



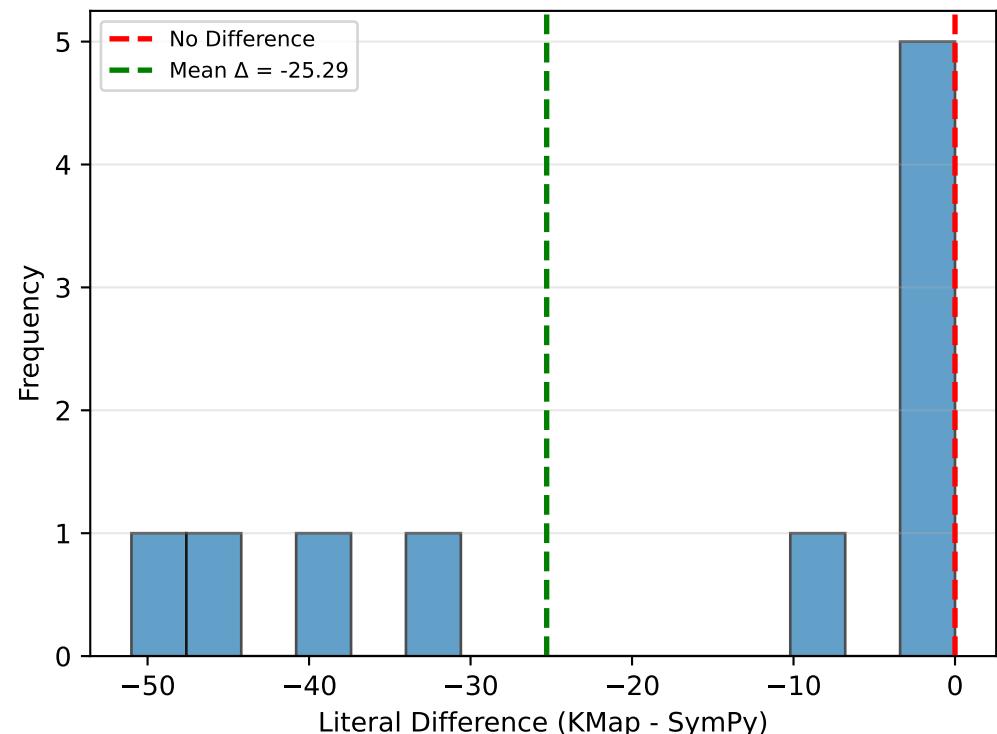
Distribution of Time Differences



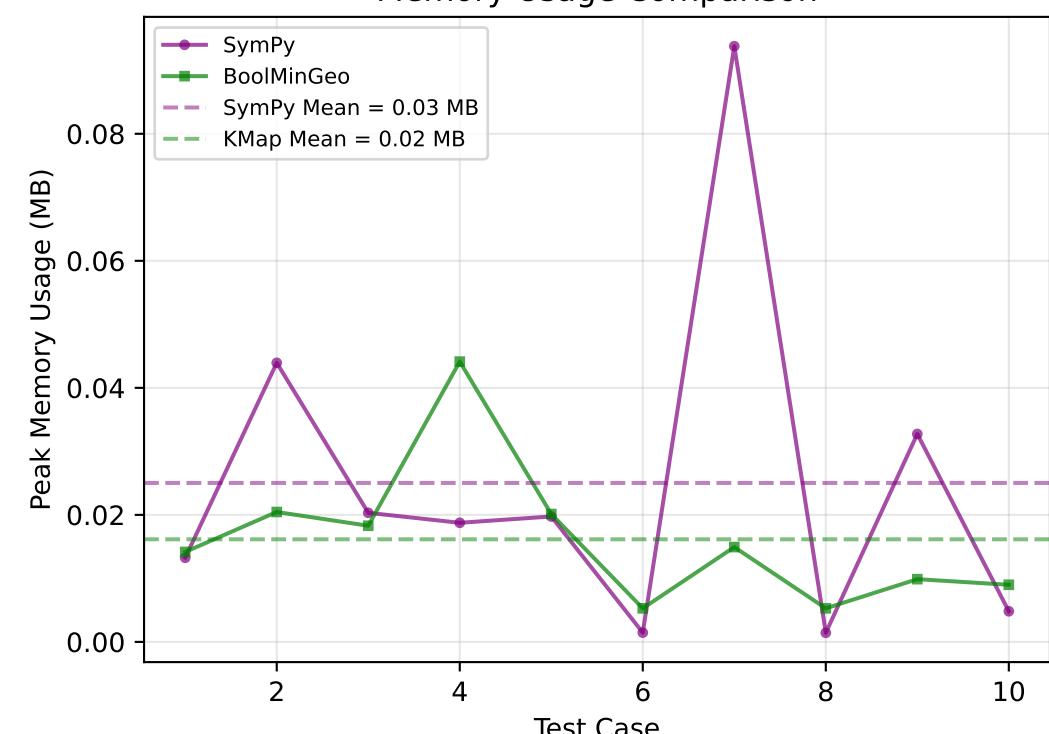
Literal Count Comparison



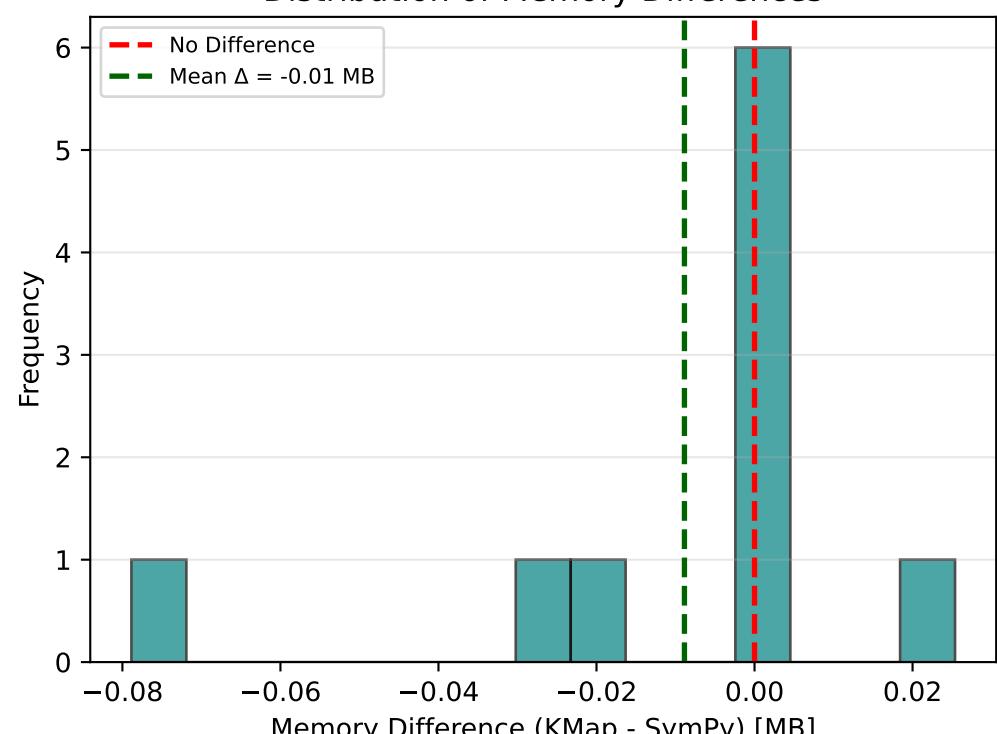
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 6-Variable K-Map (SOP Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 0.030175 s

Mean BoolMinGeo Time: 0.011805 s

Mean Difference: -0.018370 s

Std. Dev. (Δ): 0.039057 s

95% CI: [-0.046310, 0.009569]

Paired t-test: $t = -1.4874$, $p = 0.171094$

Wilcoxon test: $W = 15.0$, $p = 0.232422$

Effect Size (d): -0.4703 (small)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 35.29

Mean KMap Literals: 10.00

Mean Difference: -25.29

Std. Dev. (Δ): 21.92

95% CI: [-45.56, -5.01]

Paired t-test: $t = -3.0517$, $p = 0.022462$

Wilcoxon test: $W = 1.5$, $p = 0.062500$

Effect Size (d): -1.1534 (large)

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

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.03 MB

Mean KMap Memory: 0.02 MB

Mean Difference: -0.01 MB

Std. Dev. (Δ): 0.03 MB

95% CI: [-0.03, 0.01]

Paired t-test: $t = -0.9917$, $p = 0.347269$

Wilcoxon test: $W = 27.0$, $p = 1.000000$

Effect Size (d): -0.3136 (small)

Memory Efficiency: 1.55x

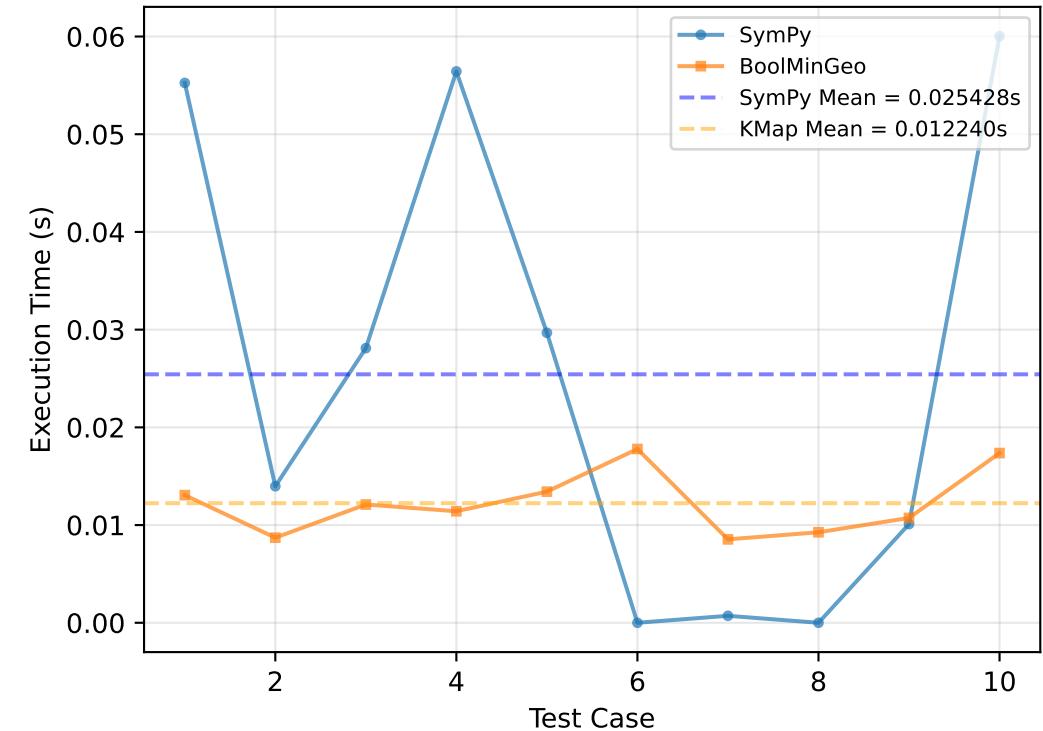
→ BoolMinGeo uses 64.6% of SymPy's memory

✗ NOT SIGNIFICANT: No significant memory difference ($p \geq 0.05$)

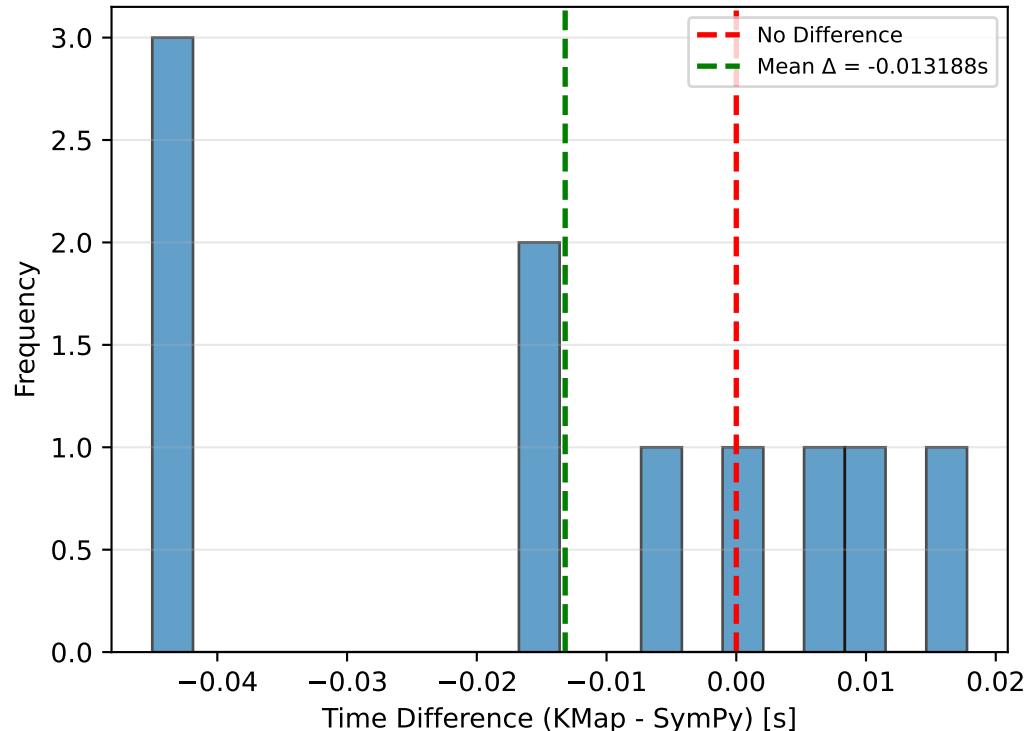
→ Both algorithms have comparable memory usage

6-Variable K-Map (POS Form)

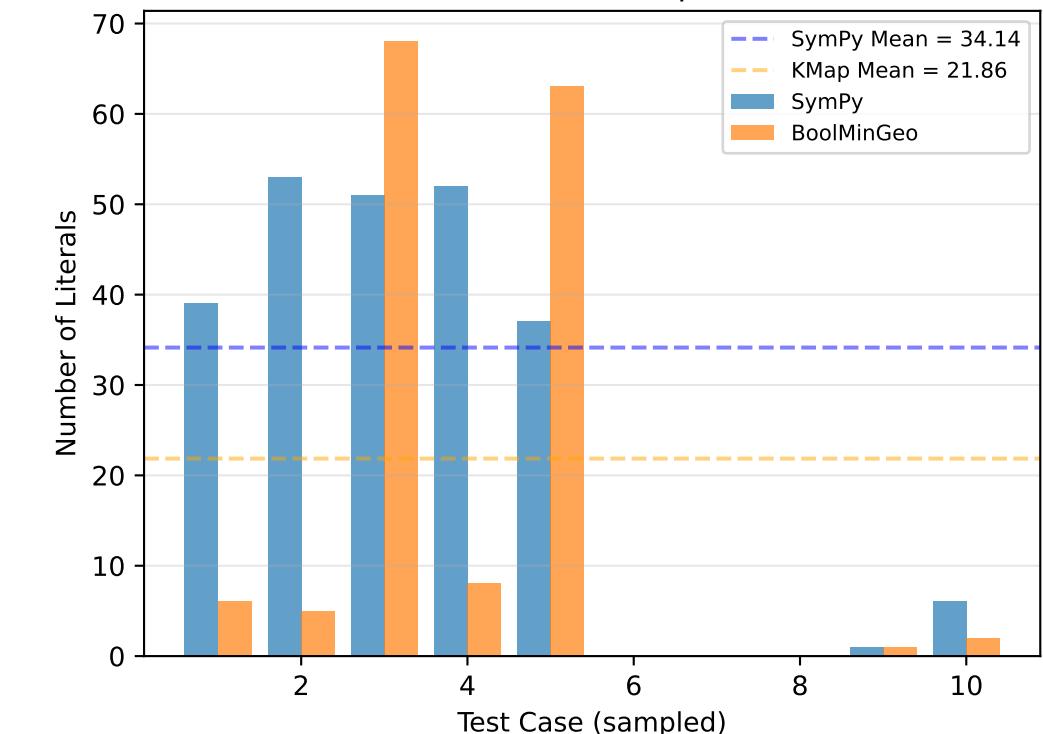
Execution Time Comparison



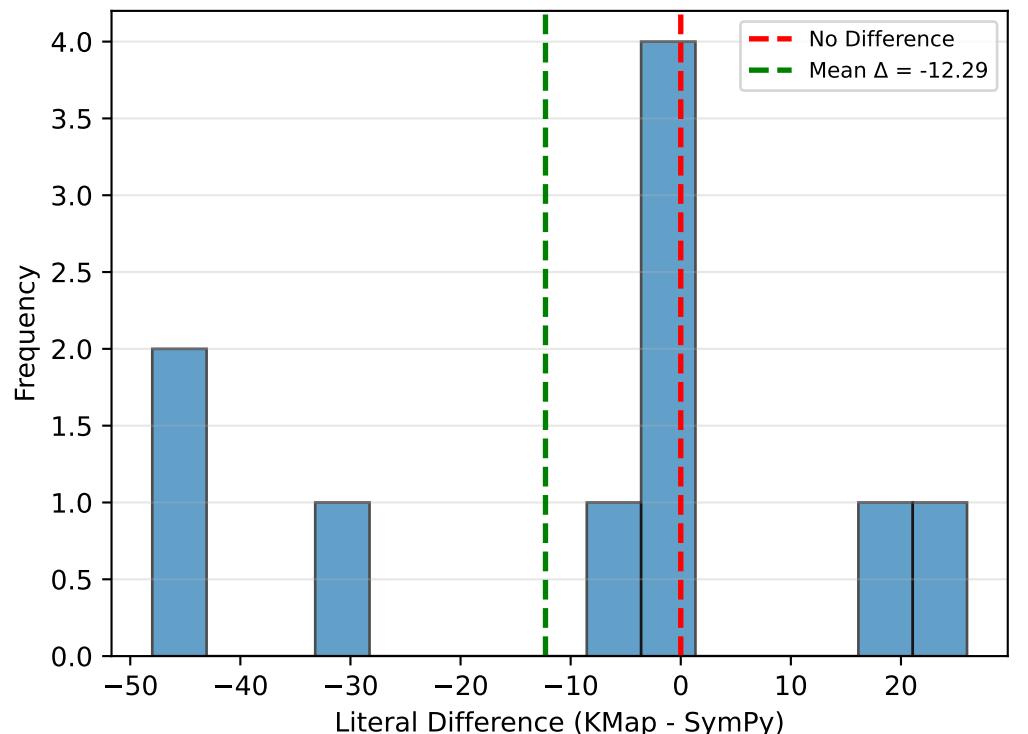
Distribution of Time Differences



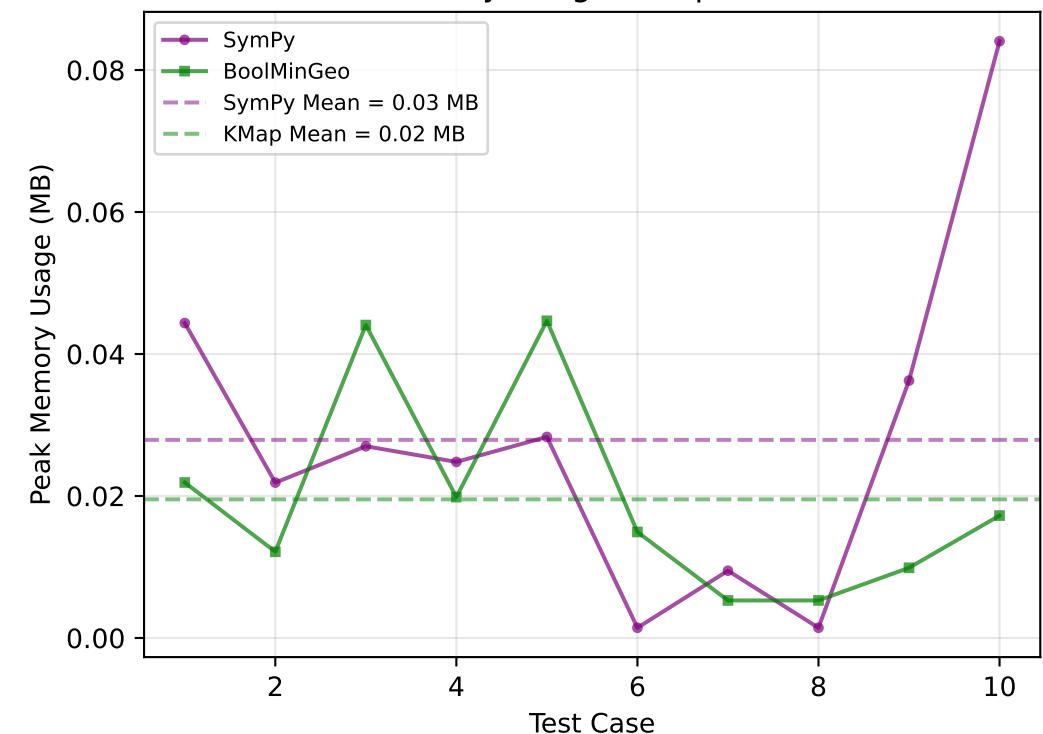
Literal Count Comparison



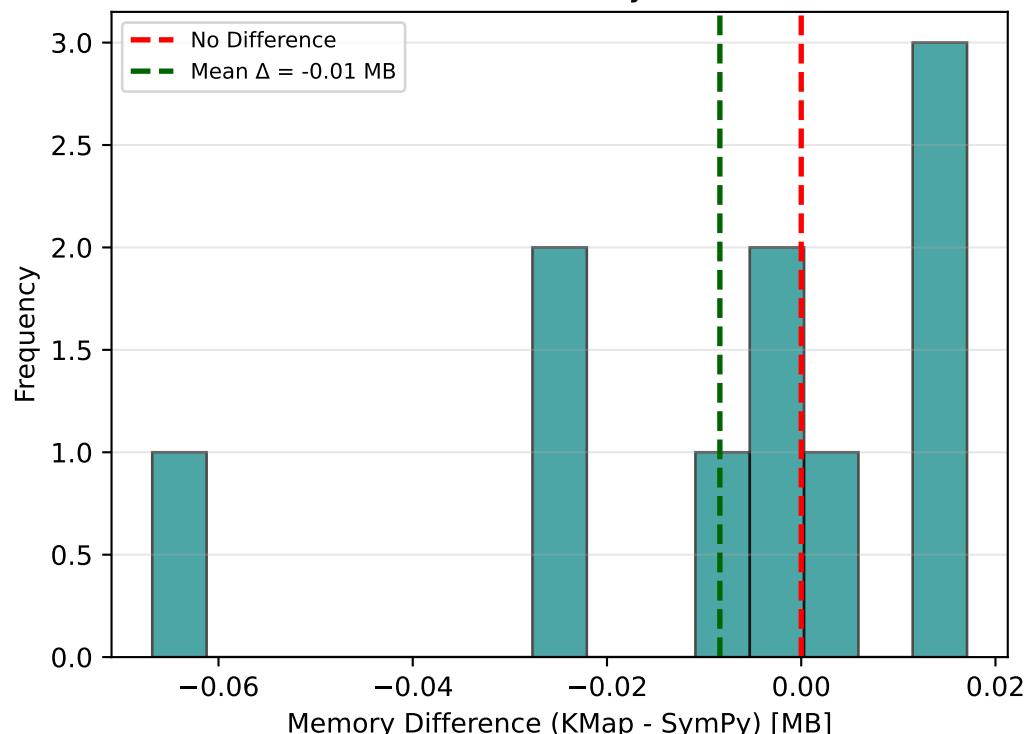
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 6-Variable K-Map (POS Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 0.025428 s

Mean BoolMinGeo Time: 0.012240 s

Mean Difference: -0.013188 s

Std. Dev. (Δ): 0.023330 s

95% CI: [-0.029877, 0.003502]

Paired t-test: $t = -1.7875$, $p = 0.107488$

Wilcoxon test: $W = 15.0$, $p = 0.232422$

Effect Size (d): -0.5653 (medium)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 34.14

Mean KMap Literals: 21.86

Mean Difference: -12.29

Std. Dev. (Δ): 29.59

95% CI: [-39.65, 15.08]

Paired t-test: $t = -1.0985$, $p = 0.314097$

Wilcoxon test: $W = 7.5$, $p = 0.343750$

Effect Size (d): -0.4152 (small)

✗ NOT SIGNIFICANT: No significant difference in simplification ($p \geq 0.05$)
→ Both algorithms achieve comparable minimization

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.03 MB

Mean KMap Memory: 0.02 MB

Mean Difference: -0.01 MB

Std. Dev. (Δ): 0.03 MB

95% CI: [-0.03, 0.01]

Paired t-test: $t = -1.0384$, $p = 0.326166$

Wilcoxon test: $W = 19.0$, $p = 0.431641$

Effect Size (d): -0.3284 (small)

Memory Efficiency: 1.43x

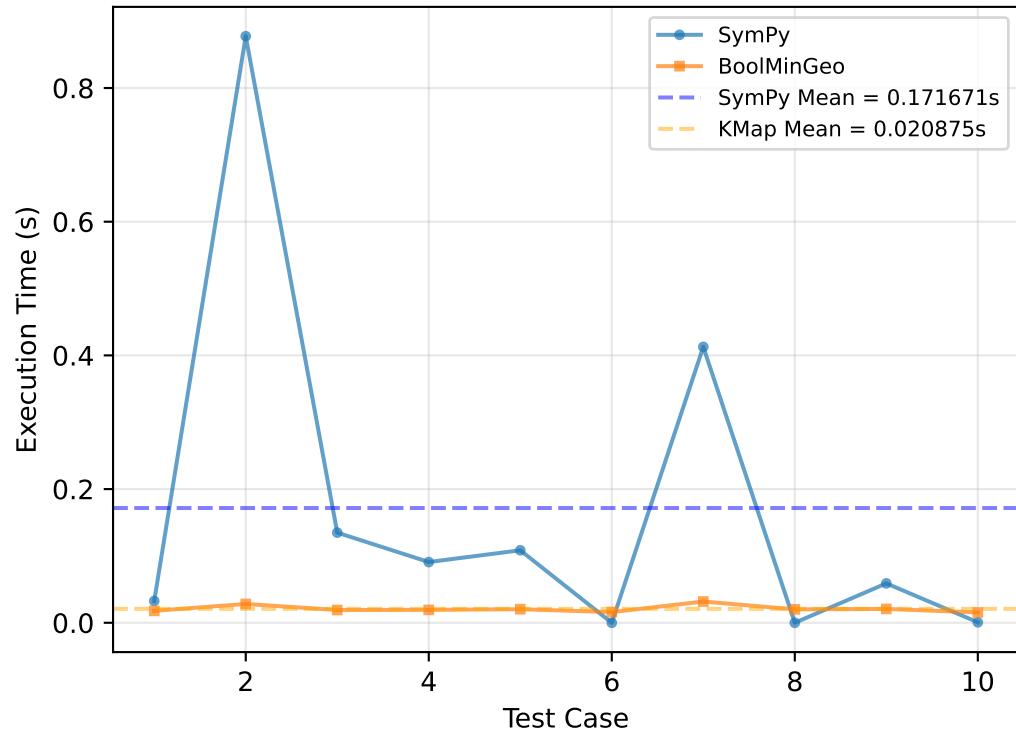
→ BoolMinGeo uses 70.0% of SymPy's memory

✗ NOT SIGNIFICANT: No significant memory difference ($p \geq 0.05$)

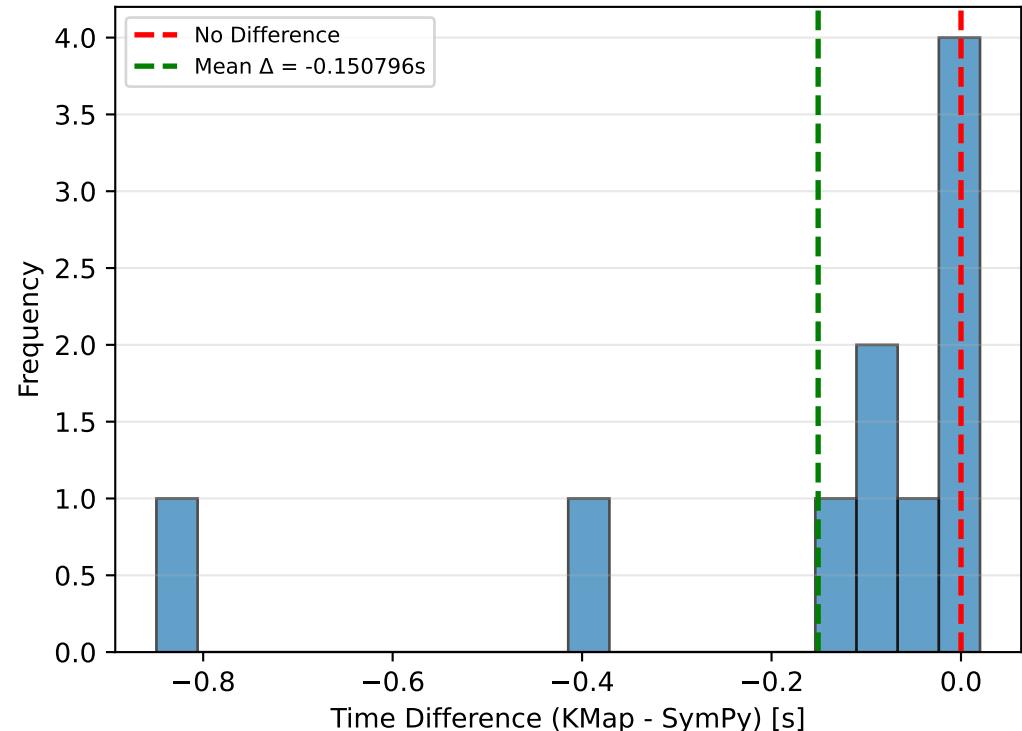
→ Both algorithms have comparable memory usage

7-Variable K-Map (SOP Form)

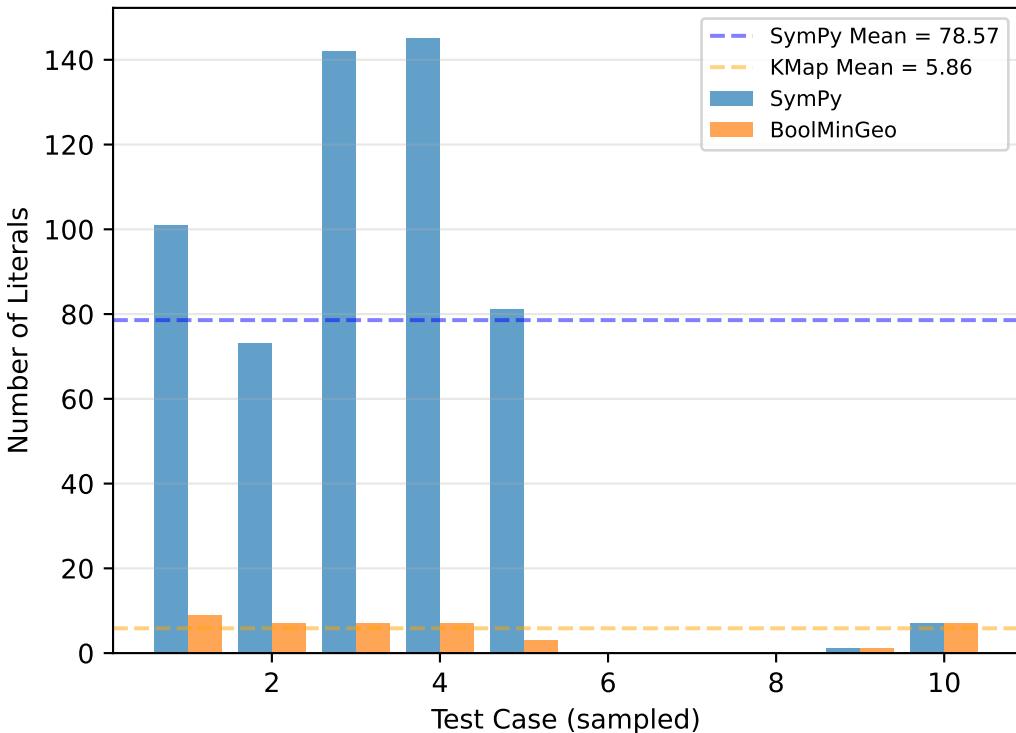
Execution Time Comparison



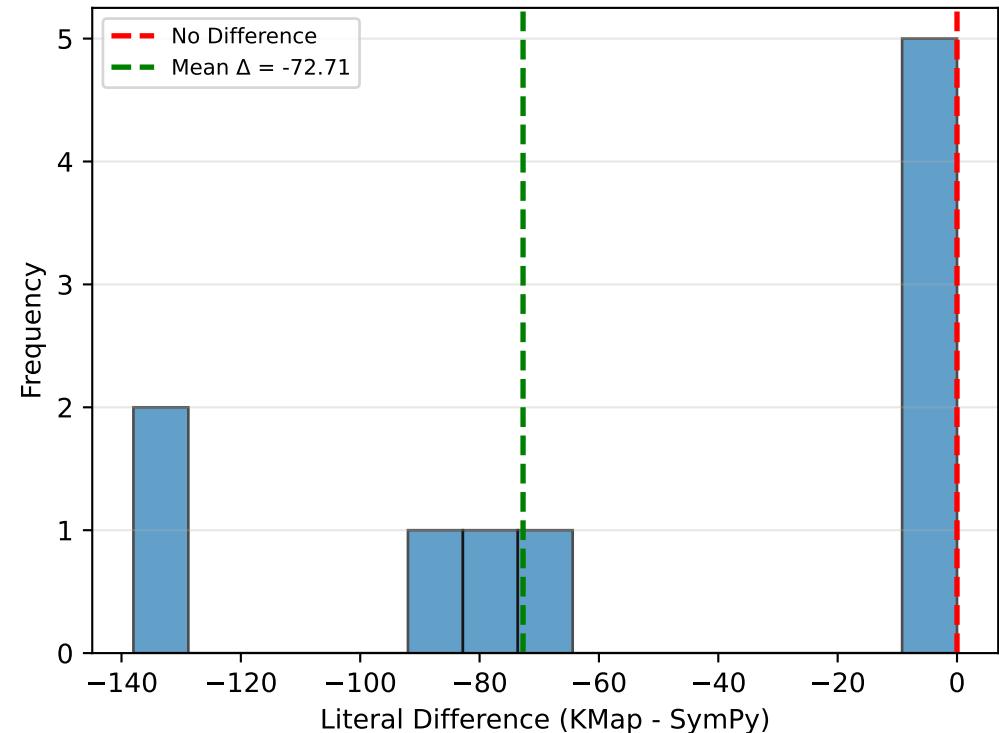
Distribution of Time Differences



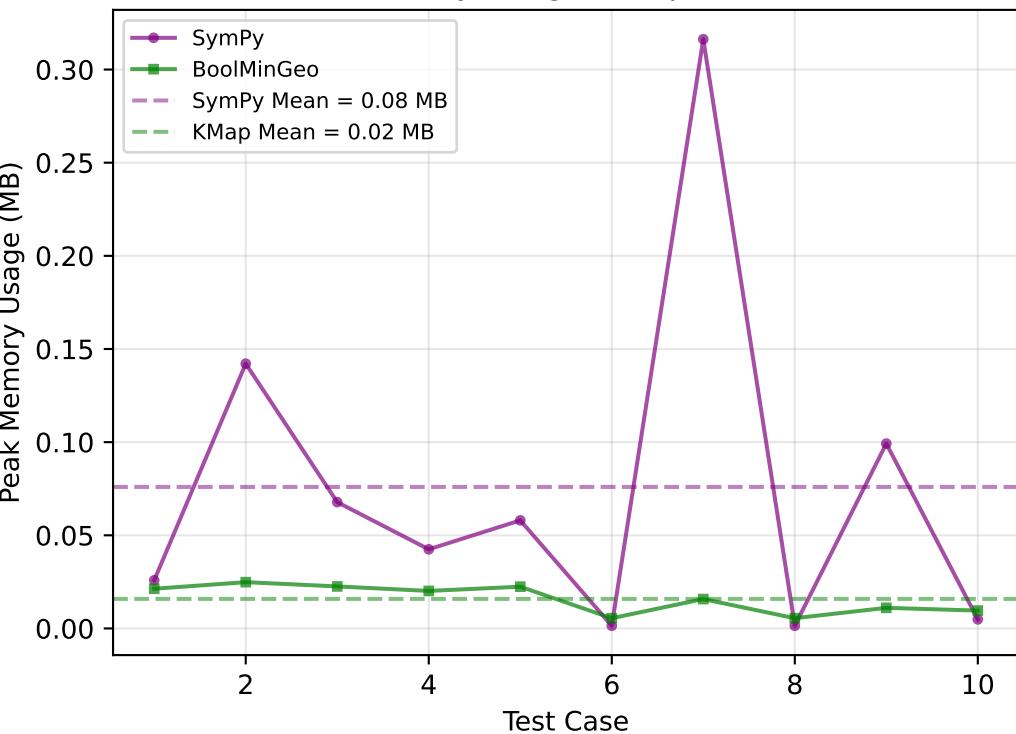
Literal Count Comparison



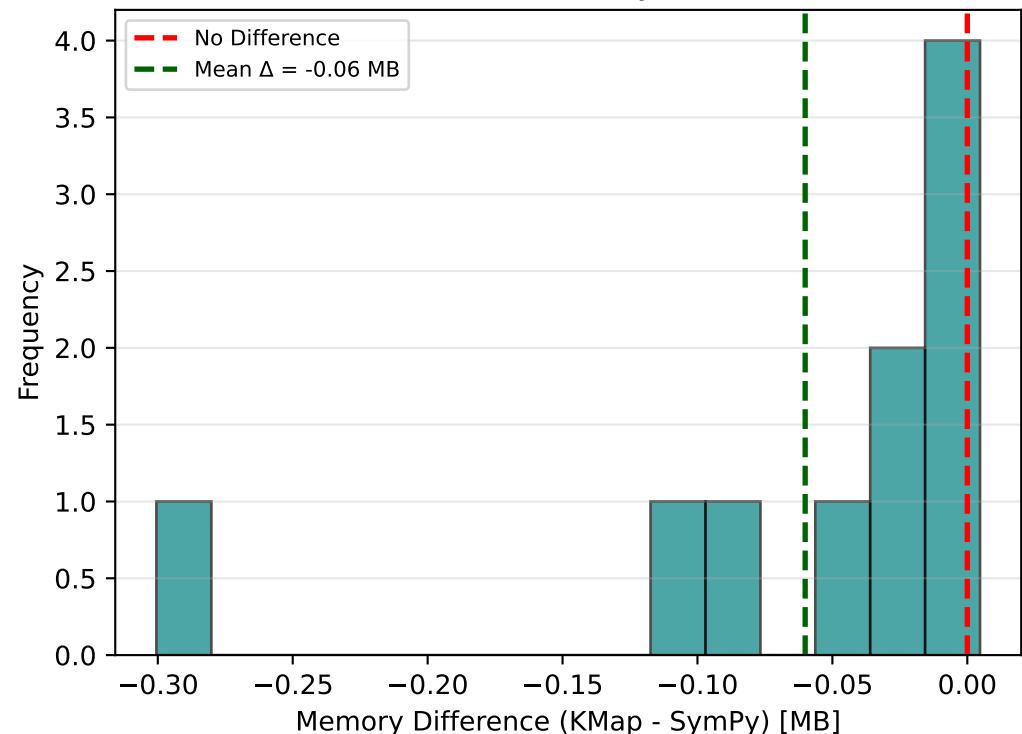
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 7-Variable K-Map (SOP Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 0.171671 s

Mean BoolMinGeo Time: 0.020875 s

Mean Difference: -0.150796 s

Std. Dev. (Δ): 0.272432 s

95% CI: [-0.345682, 0.044091]

Paired t-test: $t = -1.7504$, $p = 0.113977$

Wilcoxon test: $W = 8.0$, $p = 0.048828$

Effect Size (d): -0.5535 (medium)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 78.57

Mean KMap Literals: 5.86

Mean Difference: -72.71

Std. Dev. (Δ): 56.51

95% CI: [-124.98, -20.45]

Paired t-test: $t = -3.4043$, $p = 0.014419$

Wilcoxon test: $W = 1.5$, $p = 0.062500$

Effect Size (d): -1.2867 (large)

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

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.08 MB

Mean KMap Memory: 0.02 MB

Mean Difference: -0.06 MB

Std. Dev. (Δ): 0.09 MB

95% CI: [-0.13, 0.01]

Paired t-test: $t = -2.0201$, $p = 0.074109$

Wilcoxon test: $W = 7.0$, $p = 0.037109$

Effect Size (d): -0.6388 (medium)

Memory Efficiency: 4.79x

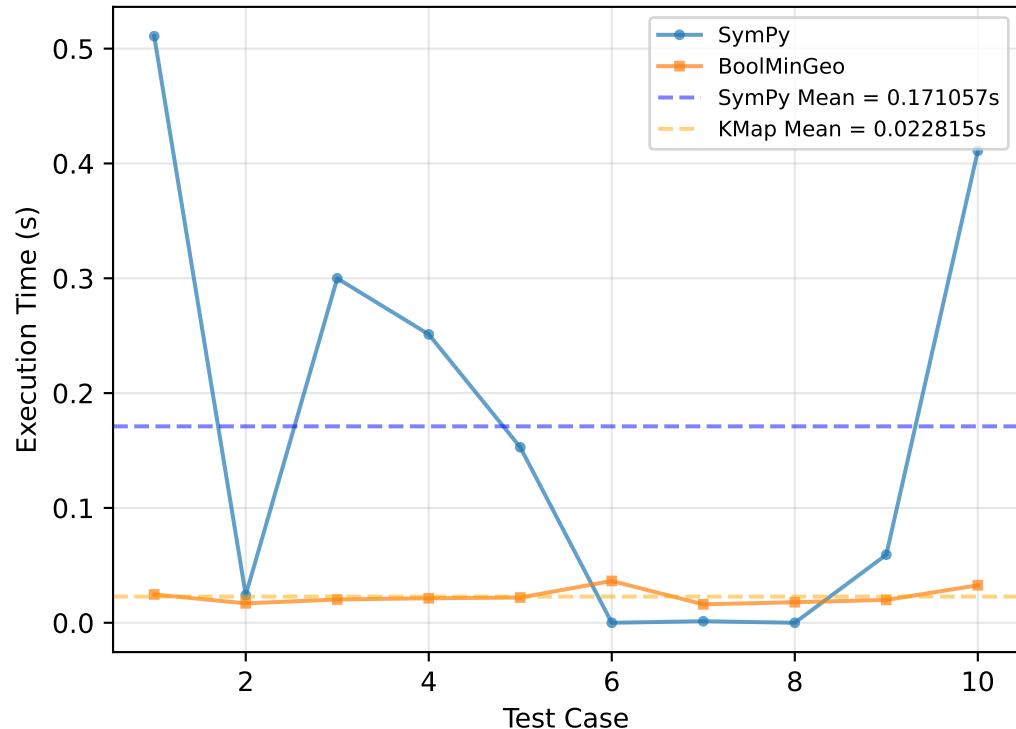
→ BoolMinGeo uses 20.9% of SymPy's memory

✗ NOT SIGNIFICANT: No significant memory difference ($p \geq 0.05$)

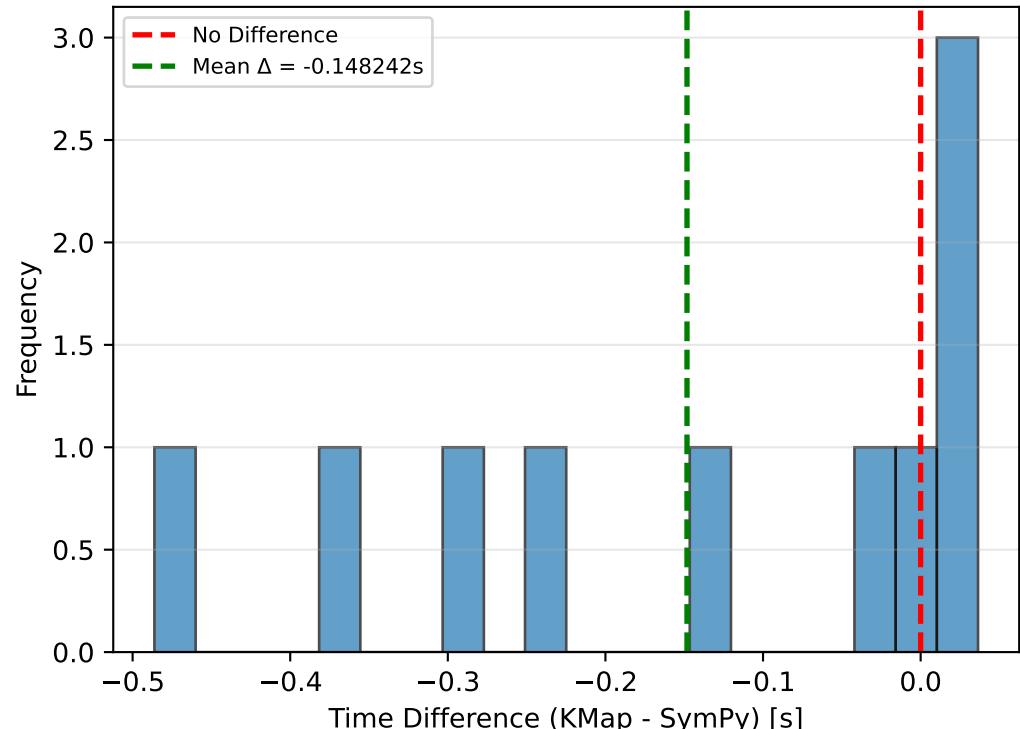
→ Both algorithms have comparable memory usage

7-Variable K-Map (POS Form)

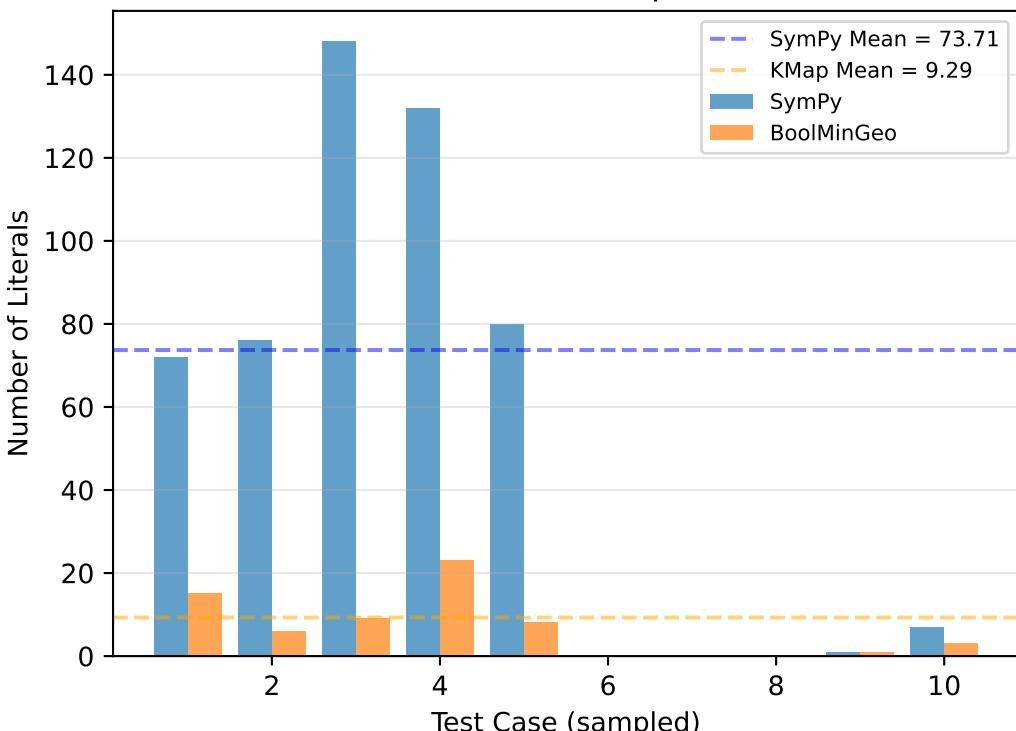
Execution Time Comparison



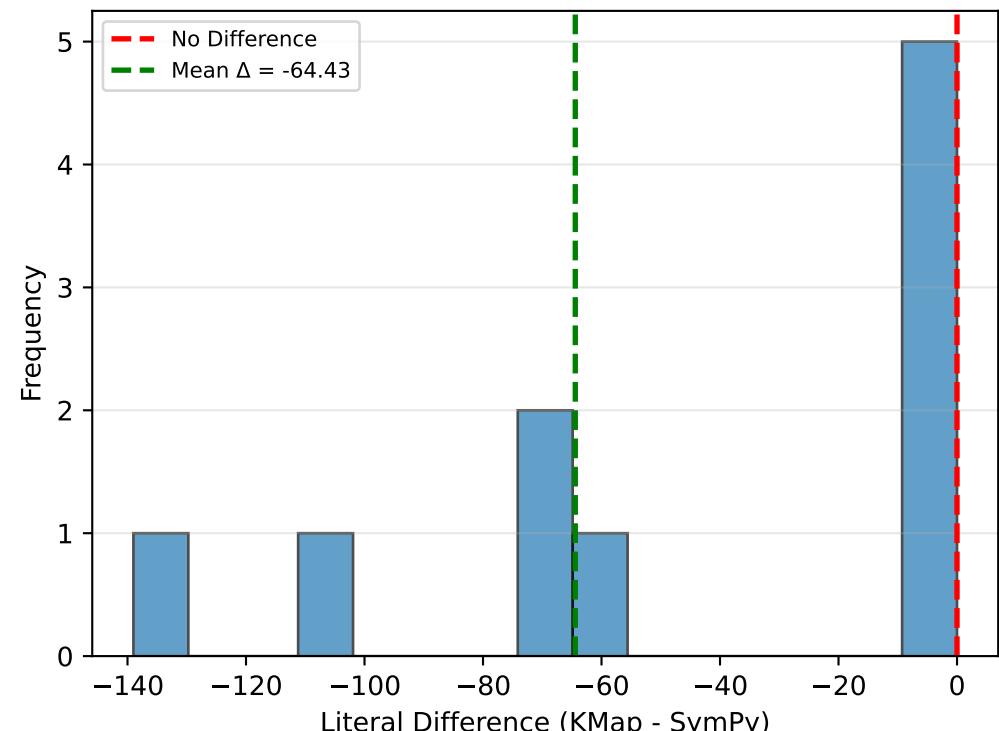
Distribution of Time Differences



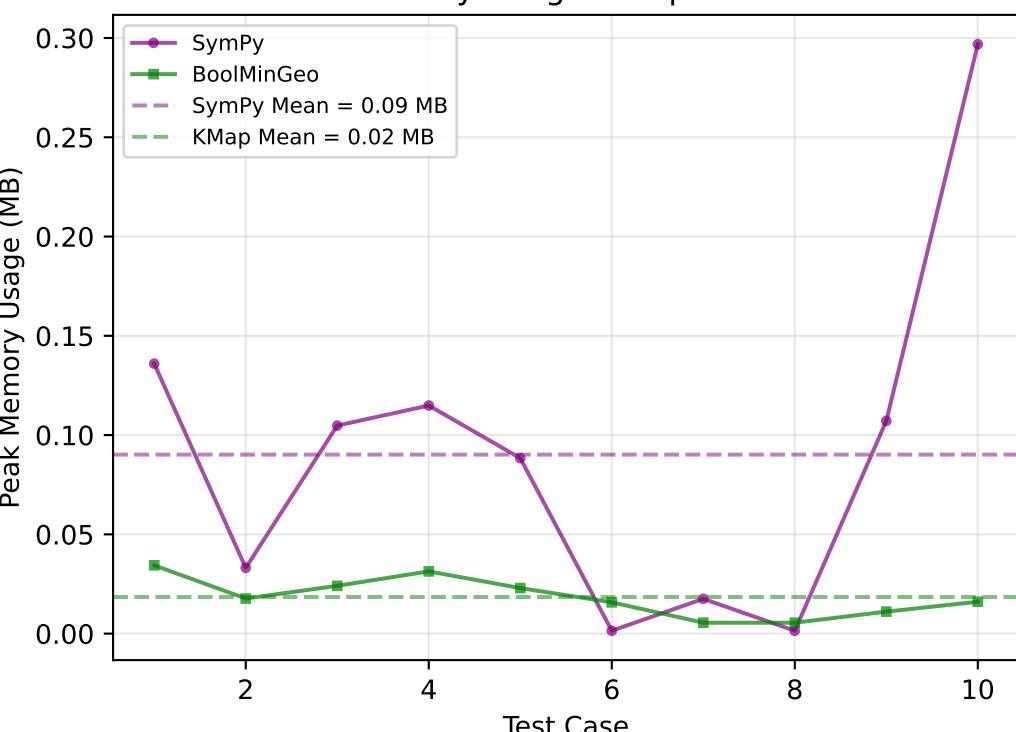
Literal Count Comparison



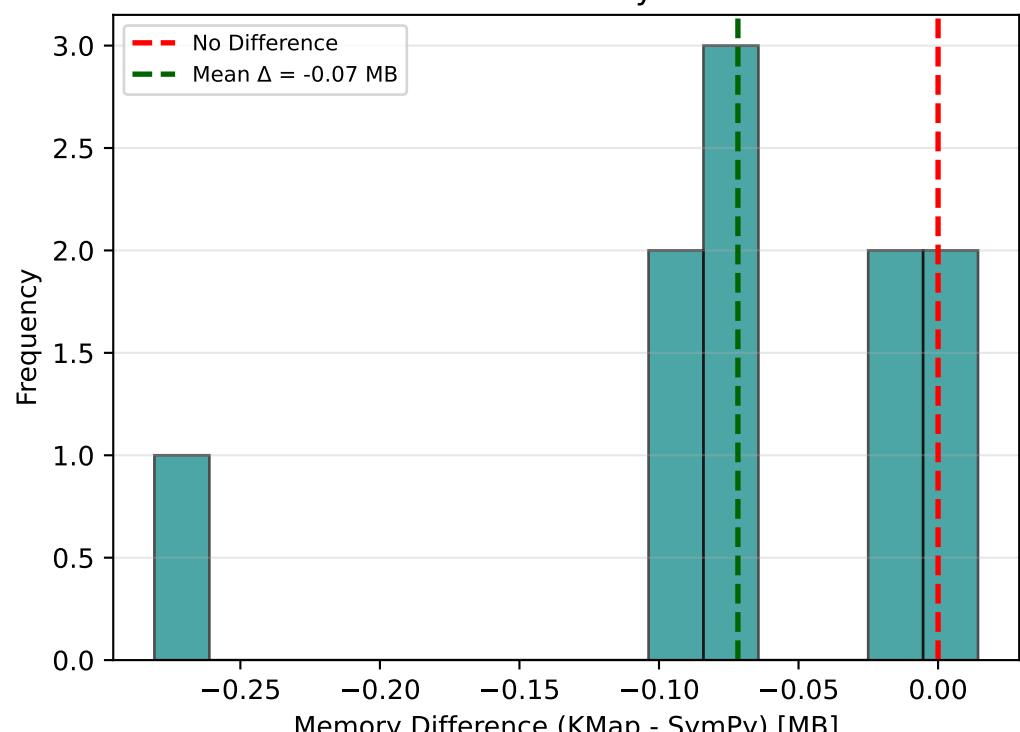
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 7-Variable K-Map (POS Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 0.171057 s

Mean BoolMinGeo Time: 0.022815 s

Mean Difference: -0.148242 s

Std. Dev. (Δ): 0.185988 s

95% CI: [-0.281290, -0.015194]

Paired t-test: $t = -2.5205$, $p = 0.032742$

Wilcoxon test: $W = 9.0$, $p = 0.064453$

Effect Size (d): -0.7971 (medium)

✓ SIGNIFICANT: Time difference is statistically significant ($p < 0.05$)
→ BoolMinGeo is significantly faster than SymPy

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 73.71

Mean KMap Literals: 9.29

Mean Difference: -64.43

Std. Dev. (Δ): 50.82

95% CI: [-111.43, -17.43]

Paired t-test: $t = -3.3545$, $p = 0.015334$

Wilcoxon test: $W = 0.5$, $p = 0.031250$

Effect Size (d): -1.2679 (large)

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

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.09 MB

Mean KMap Memory: 0.02 MB

Mean Difference: -0.07 MB

Std. Dev. (Δ): 0.09 MB

95% CI: [-0.13, -0.01]

Paired t-test: $t = -2.6620$, $p = 0.025959$

Wilcoxon test: $W = 4.0$, $p = 0.013672$

Effect Size (d): -0.8418 (large)

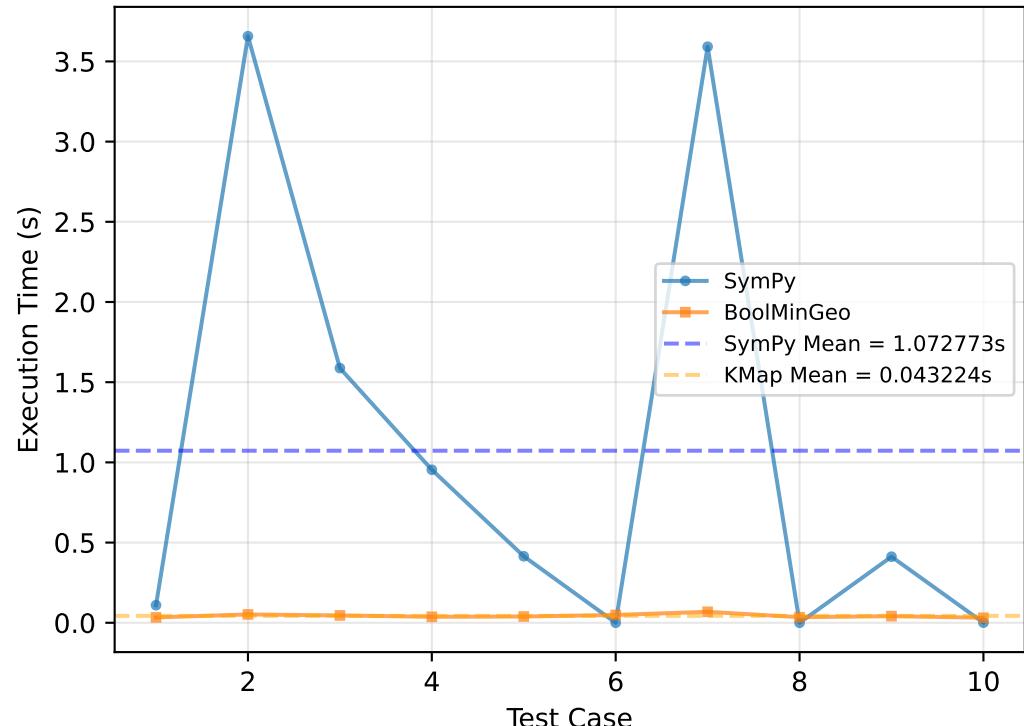
Memory Efficiency: 4.89x

→ BoolMinGeo uses 20.4% of SymPy's memory

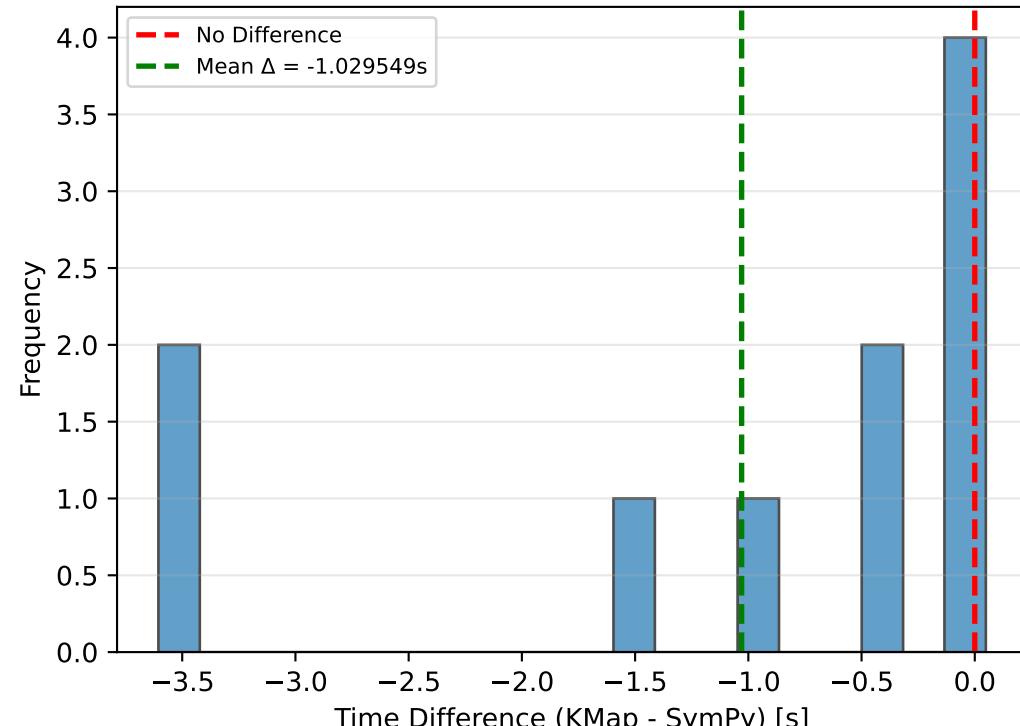
✓ SIGNIFICANT: Memory difference is statistically significant ($p < 0.05$)
→ BoolMinGeo uses significantly less memory

8-Variable K-Map (SOP Form)

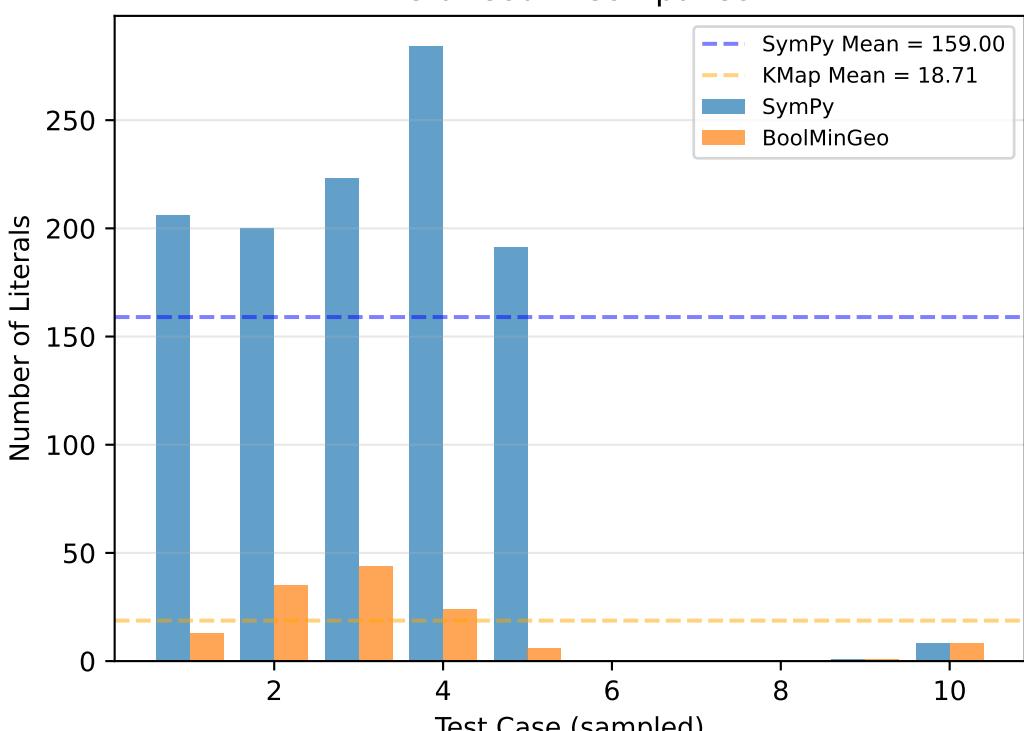
Execution Time Comparison



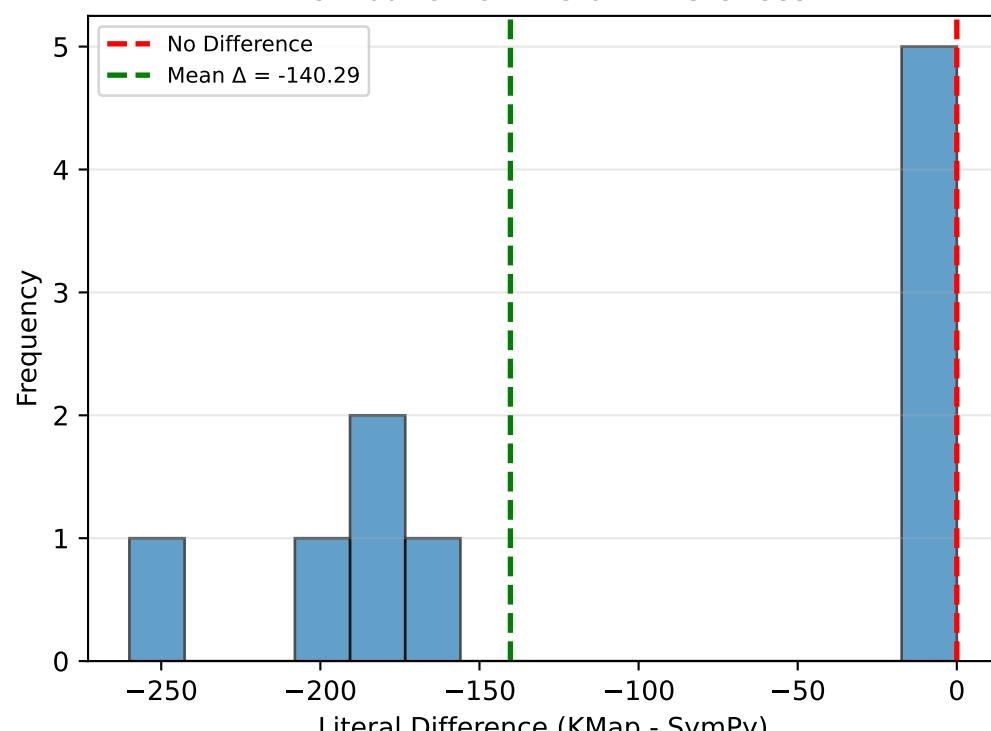
Distribution of Time Differences



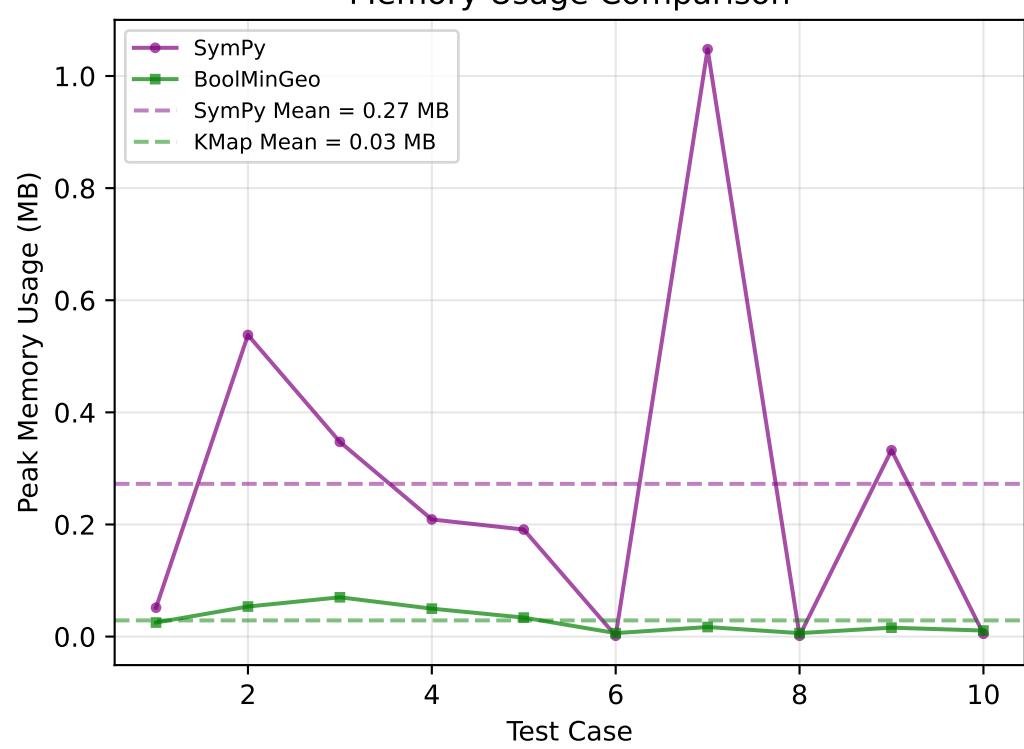
Literal Count Comparison



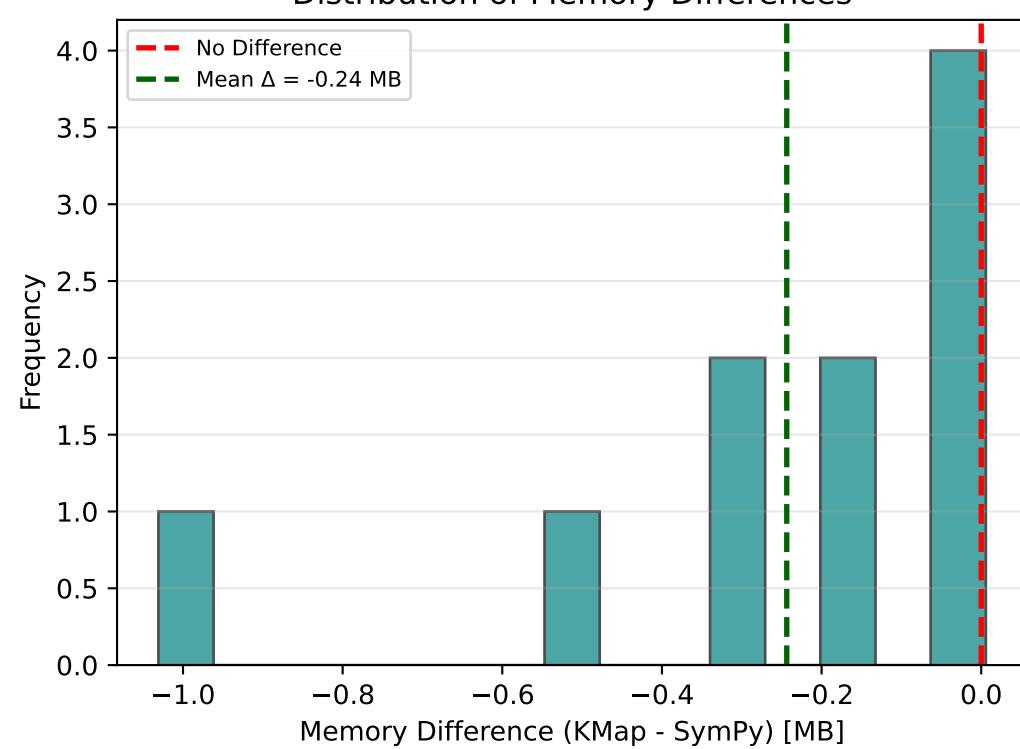
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 8-Variable K-Map (SOP Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 1.072773 s

Mean BoolMinGeo Time: 0.043224 s

Mean Difference: -1.029549 s

Std. Dev. (Δ): 1.426633 s

95% CI: [-2.050101, -0.008997]

Paired t-test: $t = -2.2821$, $p = 0.048396$

Wilcoxon test: $W = 6.0$, $p = 0.027344$

Effect Size (d): -0.7217 (medium)

✓ SIGNIFICANT: Time difference is statistically significant ($p < 0.05$)
→ BoolMinGeo is significantly faster than SymPy

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 159.00

Mean KMap Literals: 18.71

Mean Difference: -140.29

Std. Dev. (Δ): 100.48

95% CI: [-233.22, -47.36]

Paired t-test: $t = -3.6938$, $p = 0.010163$

Wilcoxon test: $W = 1.5$, $p = 0.062500$

Effect Size (d): -1.3961 (large)

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

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.27 MB

Mean KMap Memory: 0.03 MB

Mean Difference: -0.24 MB

Std. Dev. (Δ): 0.32 MB

95% CI: [-0.47, -0.01]

Paired t-test: $t = -2.3950$, $p = 0.040229$

Wilcoxon test: $W = 6.0$, $p = 0.027344$

Effect Size (d): -0.7574 (medium)

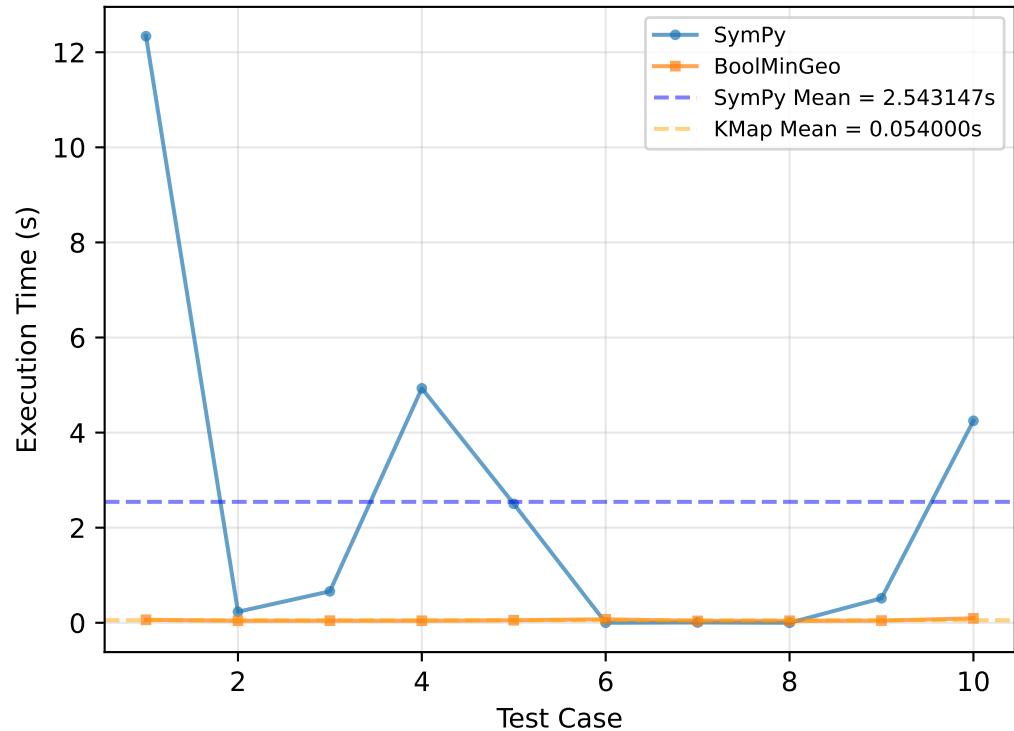
Memory Efficiency: 9.45×

→ BoolMinGeo uses 10.6% of SymPy's memory

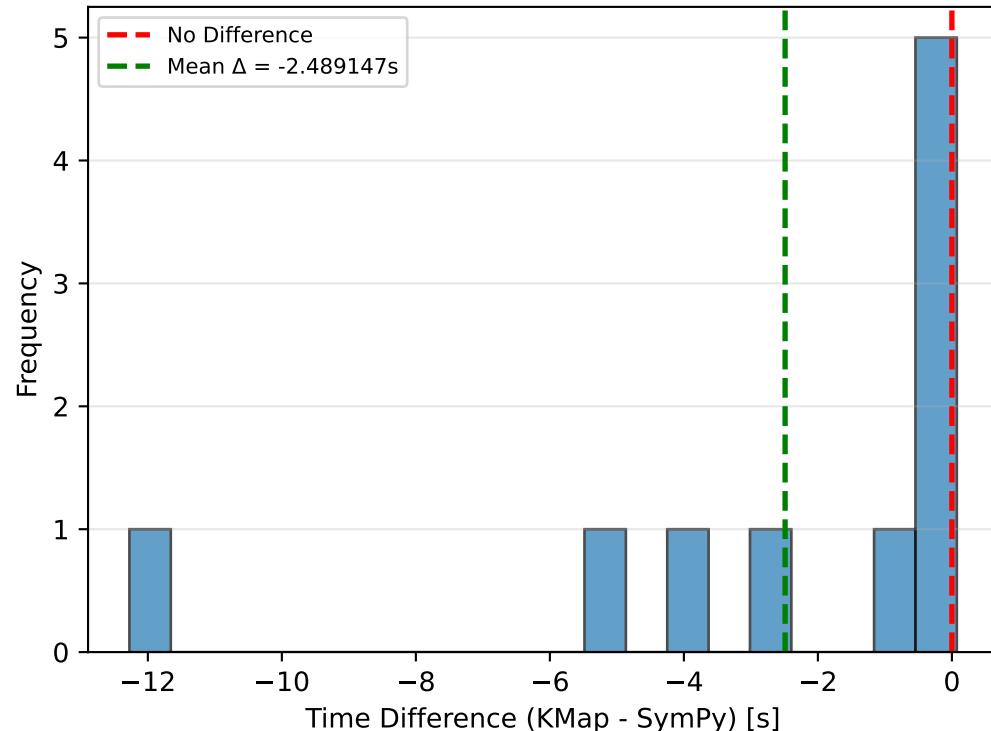
✓ SIGNIFICANT: Memory difference is statistically significant ($p < 0.05$)
→ BoolMinGeo uses significantly less memory

8-Variable K-Map (POS Form)

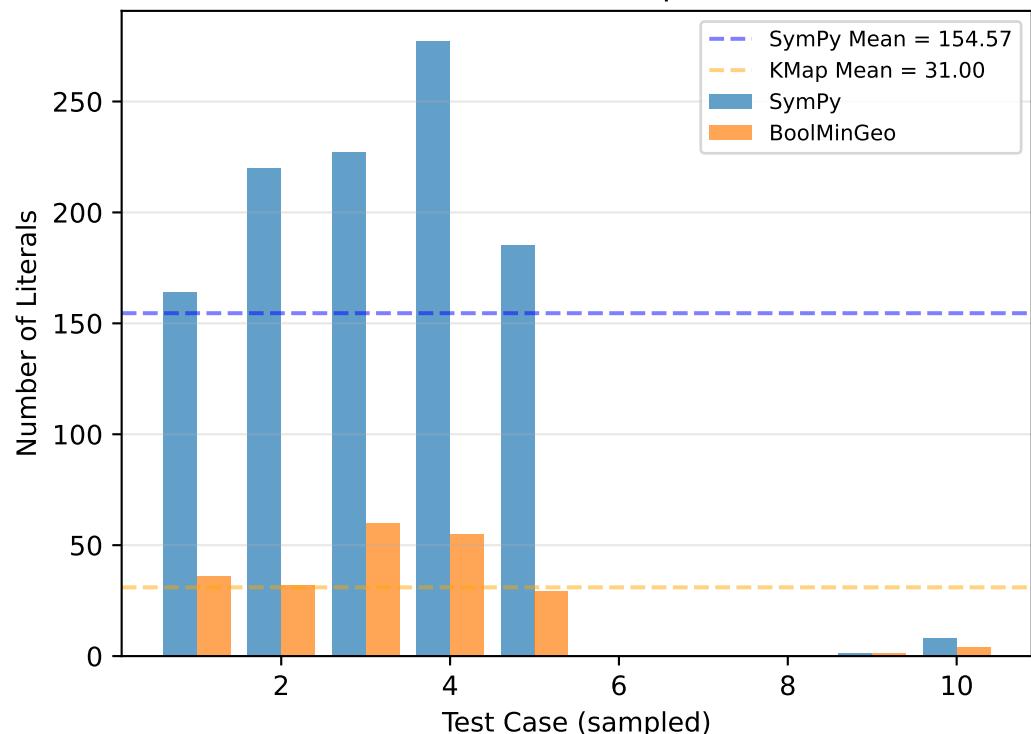
Execution Time Comparison



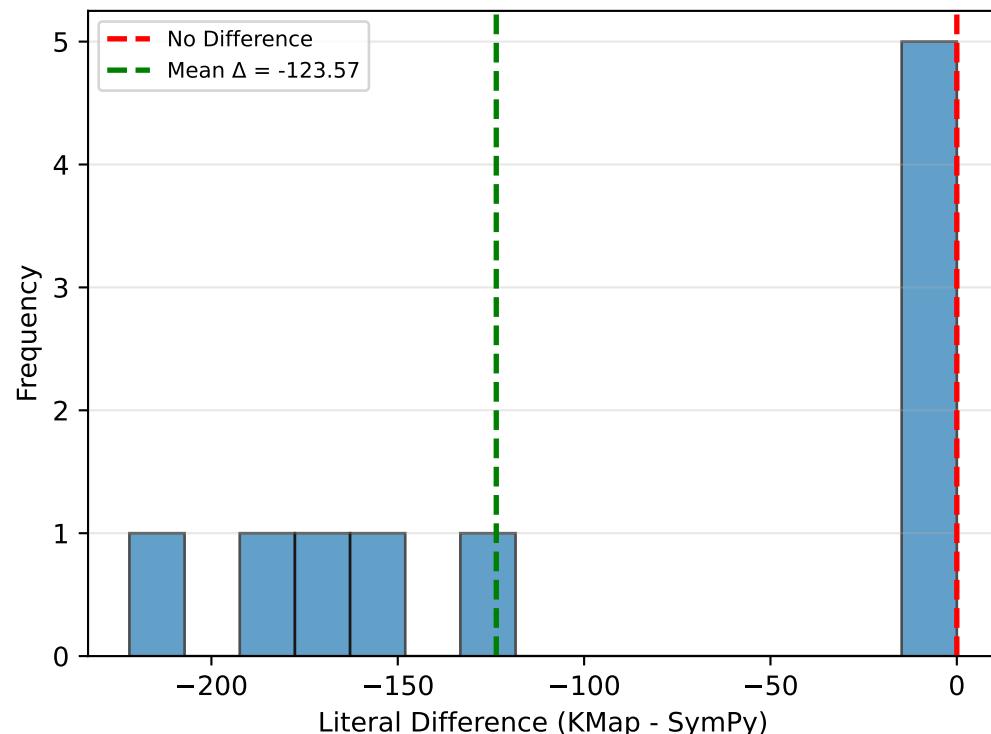
Distribution of Time Differences



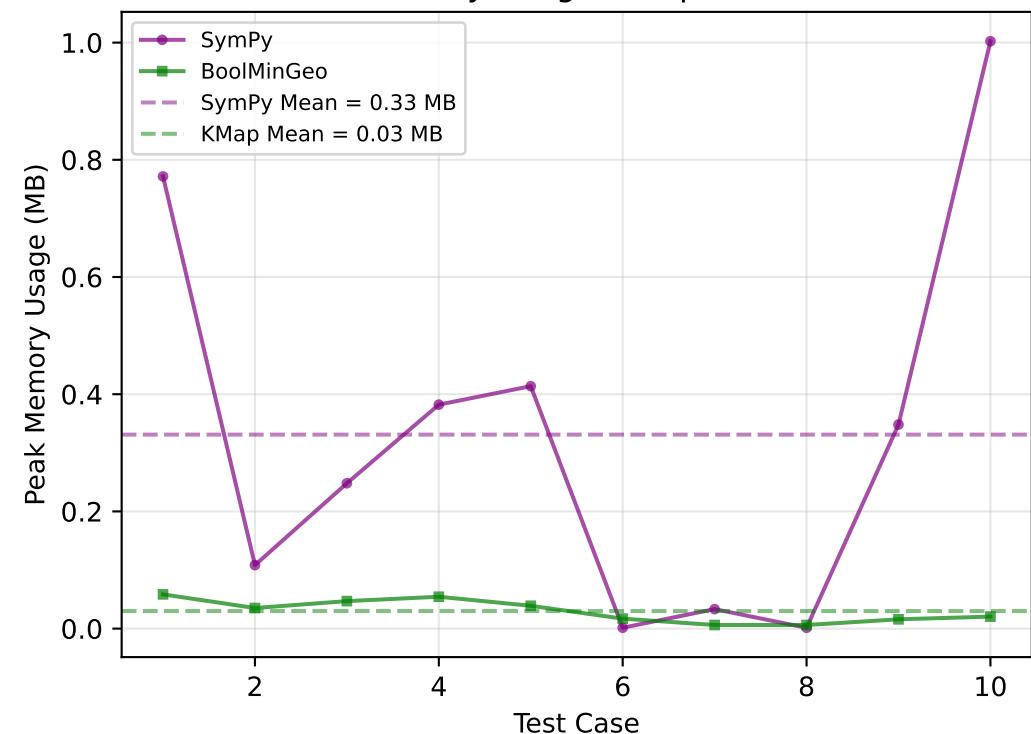
Literal Count Comparison



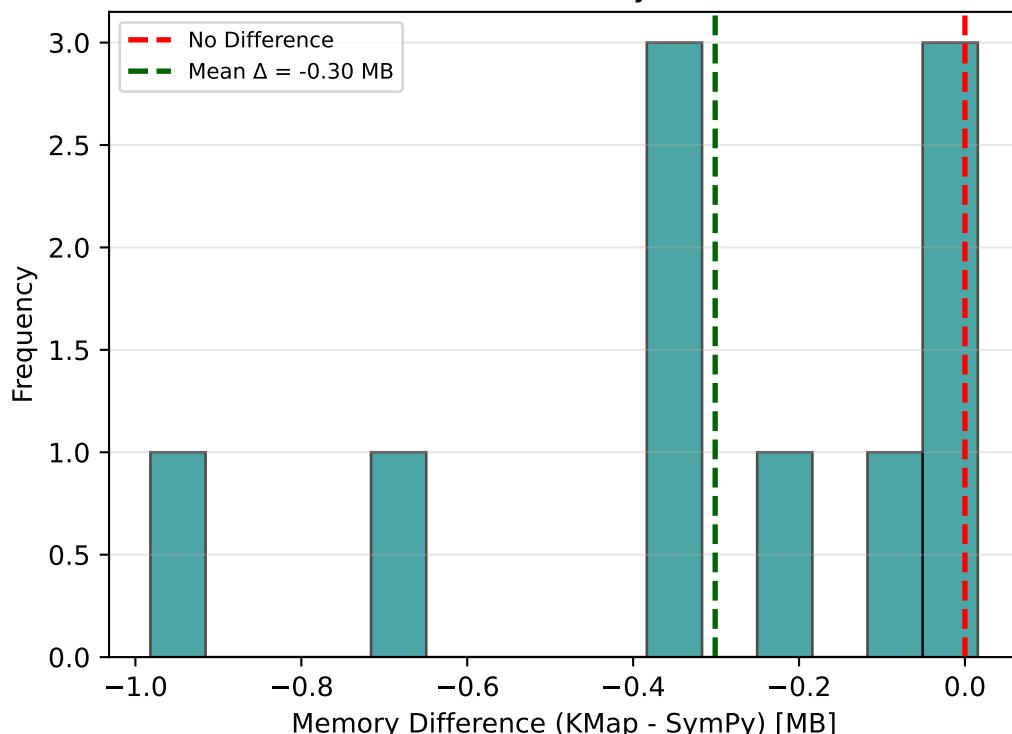
Distribution of Literal Differences



Memory Usage Comparison



Distribution of Memory Differences



STATISTICAL ANALYSIS 8-Variable K-Map (POS Form)

STATISTICAL INFERENCE REPORT

□□ TRIVIAL CONSTANT CASES DETECTED: 3/10 (30.0%)

These are degenerate constant functions (all-zeros→False, all-ones→True, all-dc) that are already maximally simplified. Both algorithms correctly identified them. Included in performance/equivalence analysis but excluded from literal-count statistics.

1. EXECUTION TIME ANALYSIS

Mean SymPy Time: 2.543147 s

Mean BoolMinGeo Time: 0.054000 s

Mean Difference: -2.489147 s

Std. Dev. (Δ): 3.894617 s

95% CI: [-5.275188, 0.296894]

Paired t-test: $t = -2.0211$, $p = 0.073992$

Wilcoxon test: $W = 6.0$, $p = 0.027344$

Effect Size (d): -0.6391 (medium)

✗ NOT SIGNIFICANT: No statistically significant time difference ($p \geq 0.05$)
→ Both algorithms exhibit comparable performance

2. SIMPLIFICATION QUALITY ANALYSIS

Analysis based on 7 non-constant functions:

(3 constant function(s) excluded from this analysis)

Mean SymPy Literals: 154.57

Mean KMap Literals: 31.00

Mean Difference: -123.57

Std. Dev. (Δ): 87.90

95% CI: [-204.87, -42.27]

Paired t-test: $t = -3.7192$, $p = 0.009861$

Wilcoxon test: $W = 0.5$, $p = 0.031250$

Effect Size (d): -1.4057 (large)

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

3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

Mean SymPy Memory: 0.33 MB

Mean KMap Memory: 0.03 MB

Mean Difference: -0.30 MB

Std. Dev. (Δ): 0.33 MB

95% CI: [-0.54, -0.07]

Paired t-test: $t = -2.8985$, $p = 0.017639$

Wilcoxon test: $W = 3.0$, $p = 0.009766$

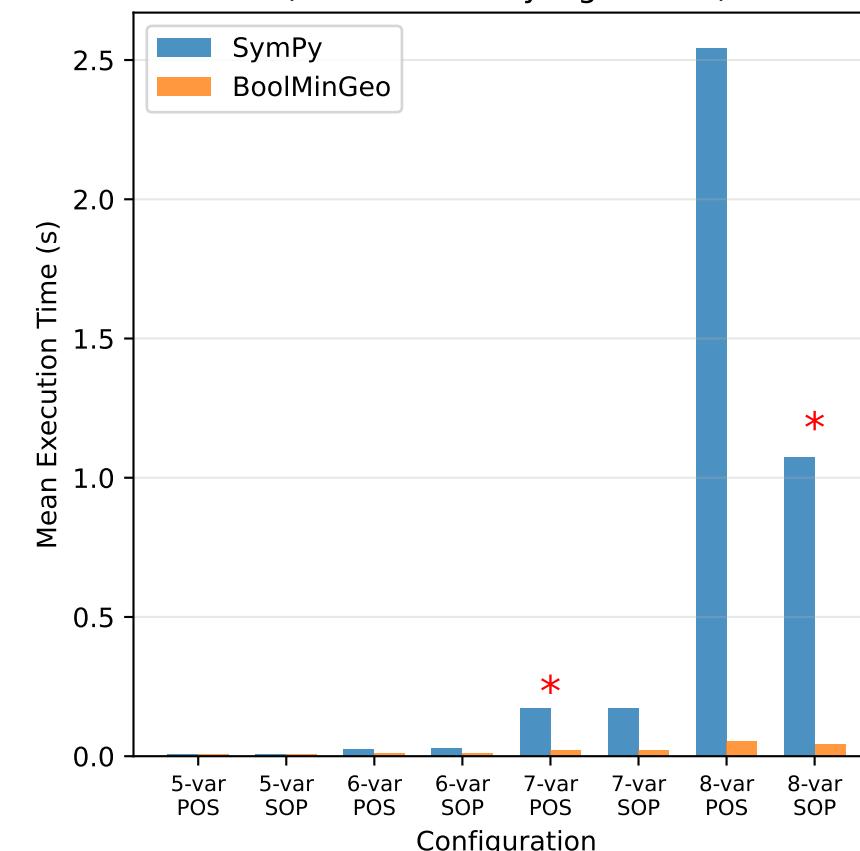
Effect Size (d): -0.9166 (large)

Memory Efficiency: 11.05×

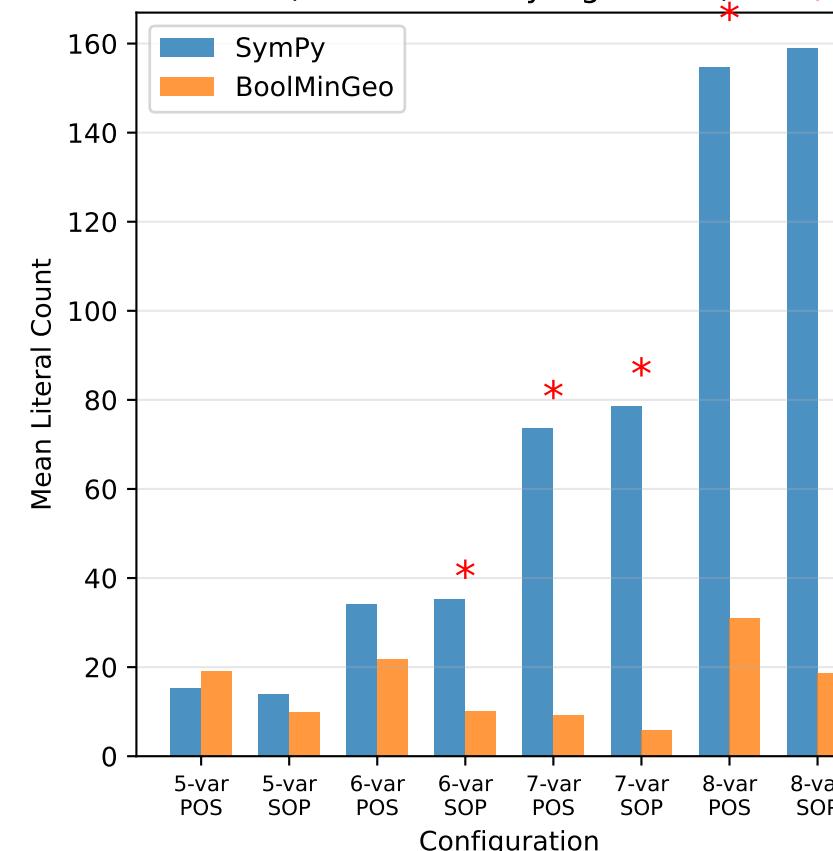
→ BoolMinGeo uses 9.1% of SymPy's memory

✓ SIGNIFICANT: Memory difference is statistically significant ($p < 0.05$)
→ BoolMinGeo uses significantly less memory

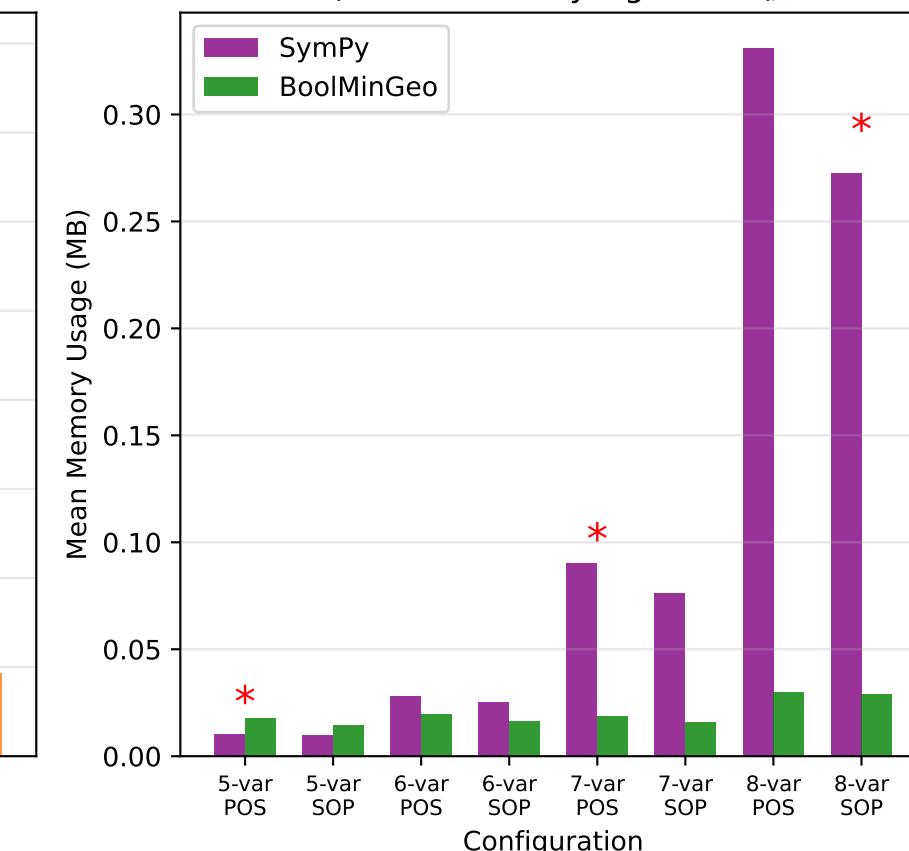
Time Performance by Configuration
(* = statistically significant)



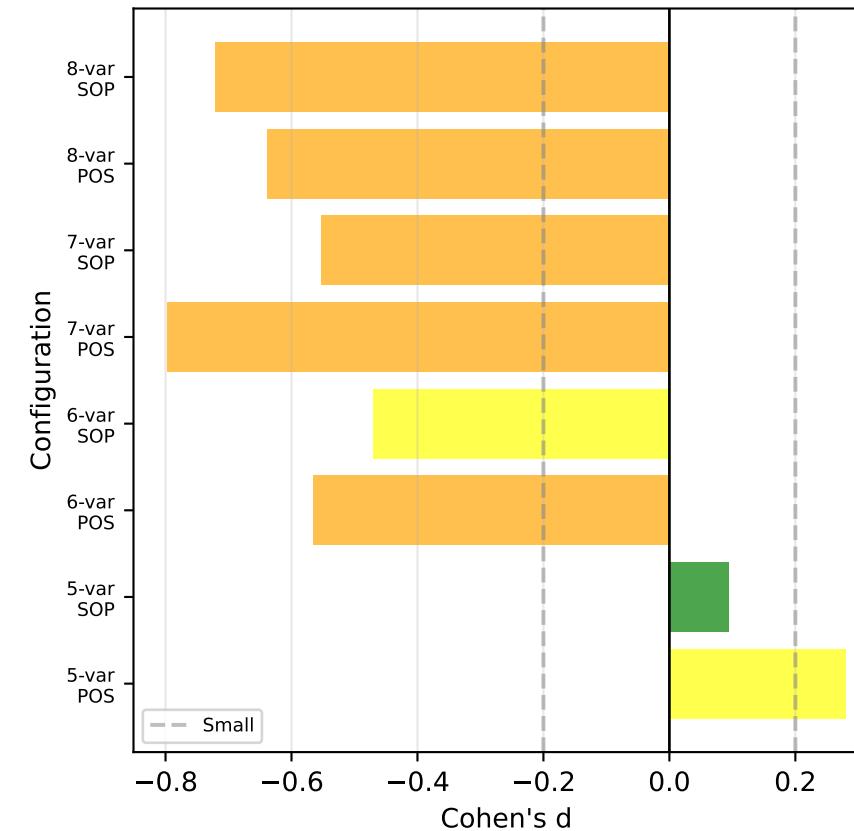
Average Simplification Quality
(* = statistically significant)



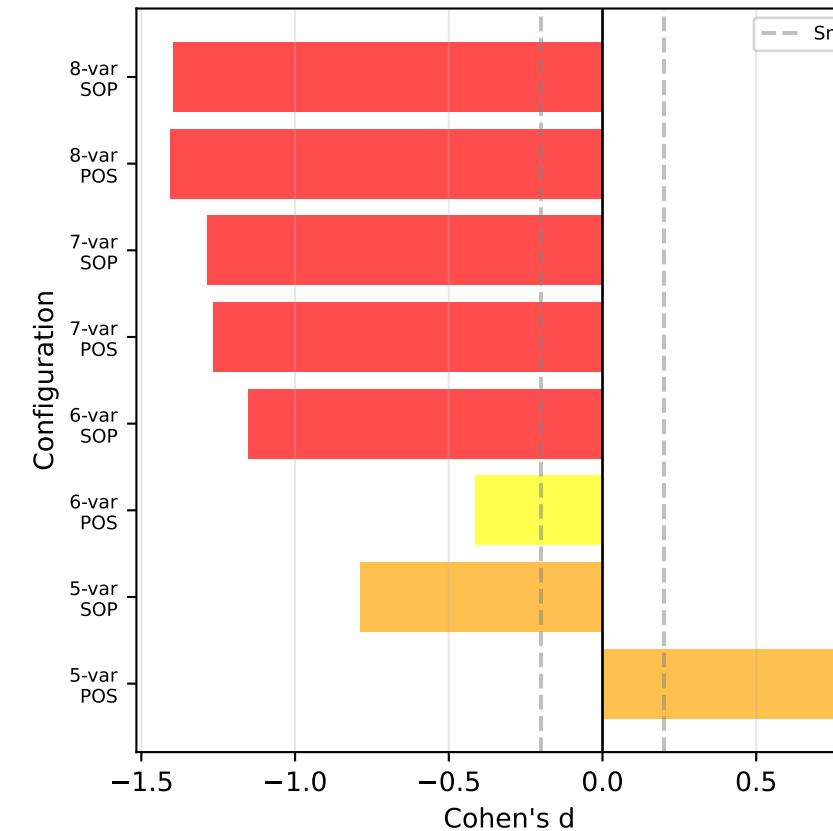
Memory Usage by Configuration
(* = statistically significant)



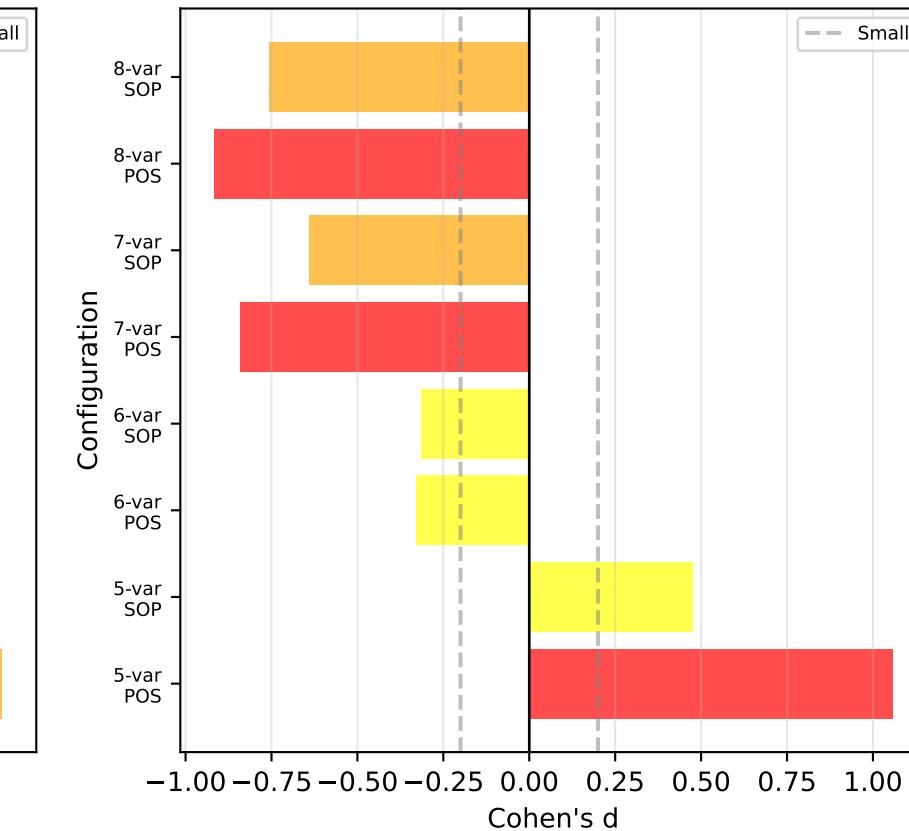
Effect Size: Time
(Negative = KMap faster)



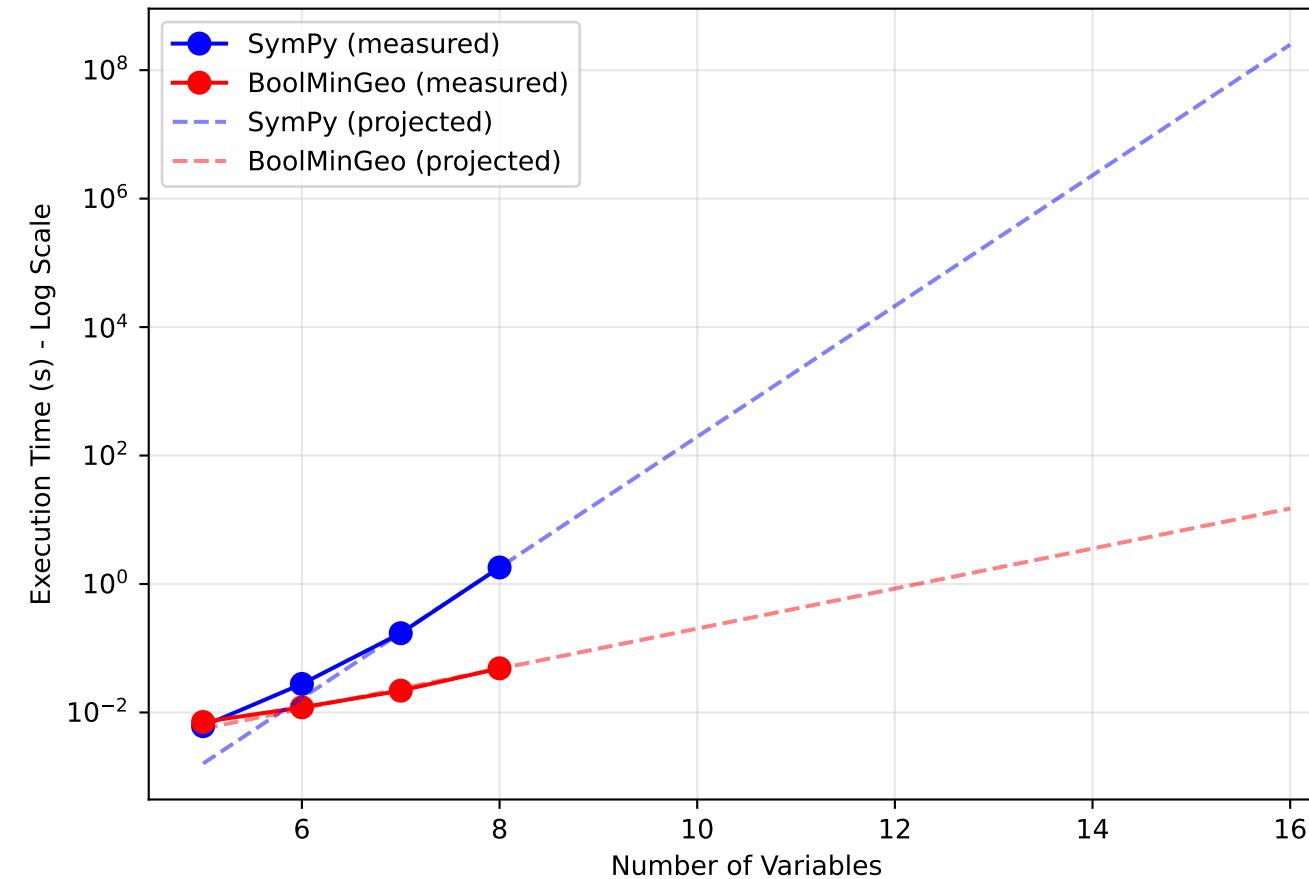
Effect Size: Literals
(Negative = KMap minimal)



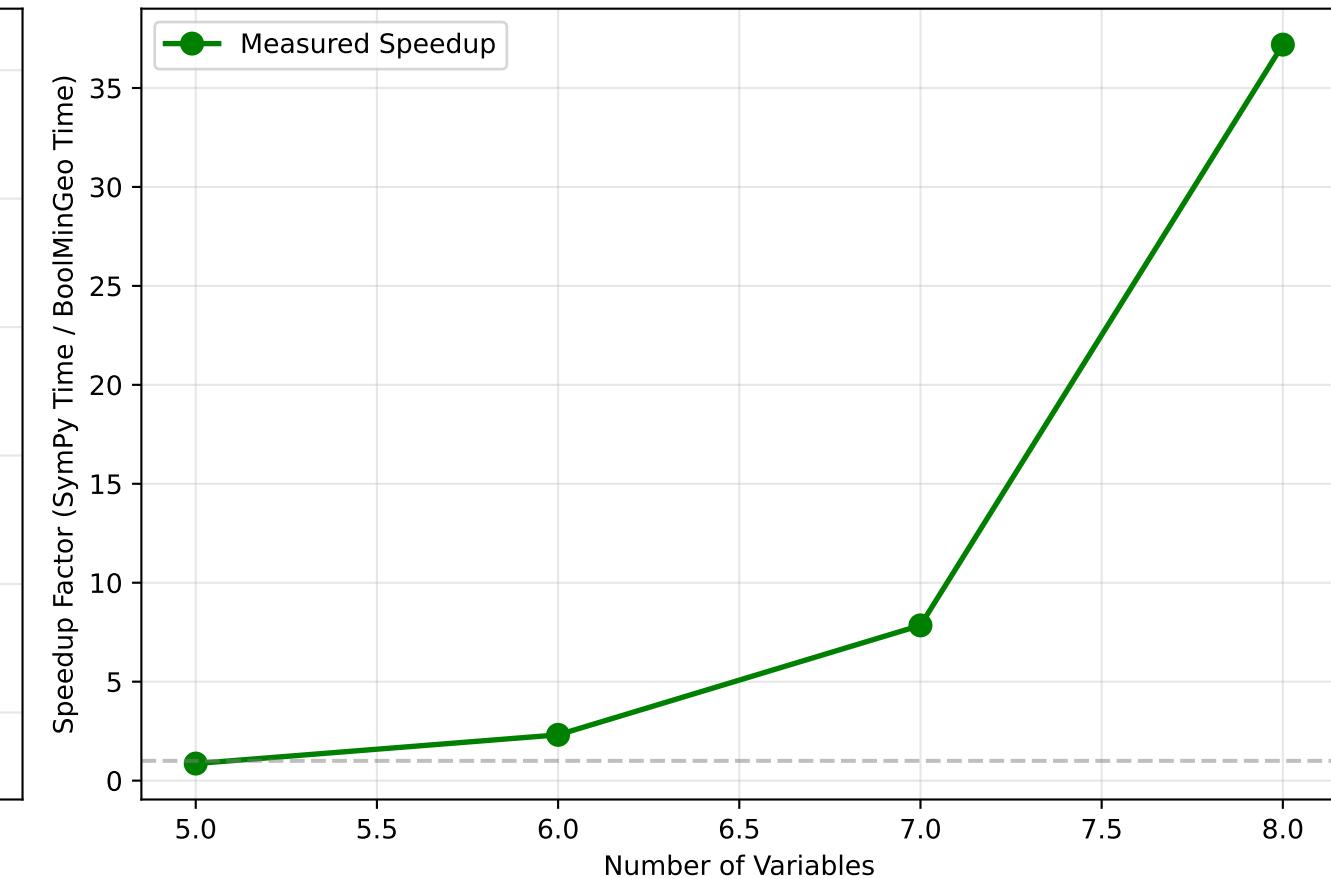
Effect Size: Memory
(Negative = KMap efficient)



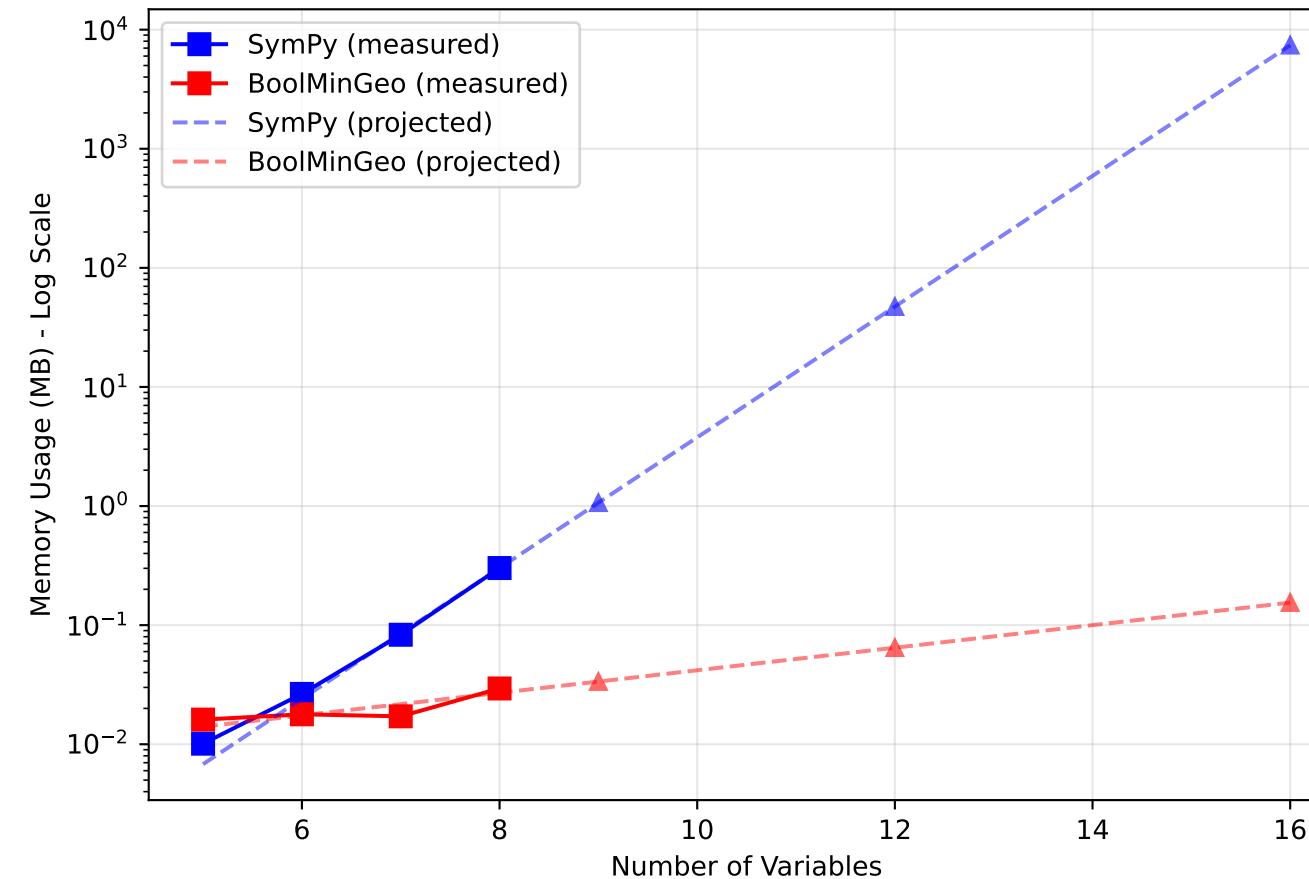
Scalability: Execution Time vs Problem Size



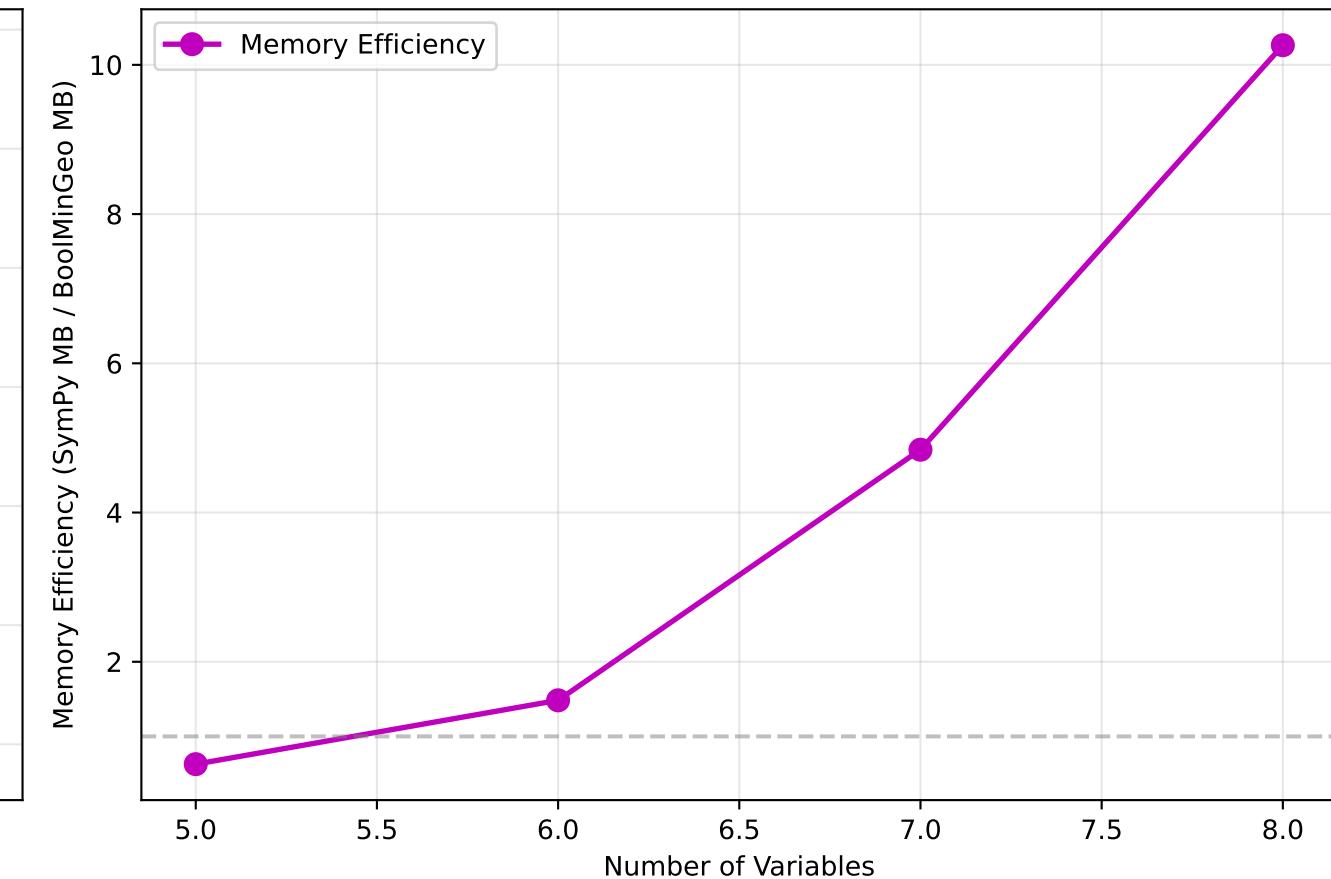
Time Efficiency: BoolMinGeo Speedup



Space Complexity: Memory vs Problem Size



Space Efficiency: Relative Memory Usage



SCALABILITY ANALYSIS

COMPLEXITY MODELS

SymPy Exponential Model:

$$T \approx 1.31e-08 \times 10.412^n$$

BoolMinGeo Exponential Model:

$$T \approx 1.54e-04 \times 2.050^n$$

Growth Rate Analysis:

SymPy base growth factor: 10.412

BoolMinGeo base growth factor: 2.050

Ratio (SymPy/KMap): 5.08x

→ SymPy's execution time grows 5.08x faster per additional variable compared to BoolMinGeo

MODEL VALIDATION

Prediction accuracy (measured vs model):

5-var: SymPy 73.9% error, KMap 21.3% error

6-var: SymPy 40.0% error, KMap 4.7% error

7-var: SymPy 1.3% error, KMap 7.5% error

8-var: SymPy 0.0% error, KMap 1.0% error

Model fit quality: Acceptable

OBSERVED PERFORMANCE

Measured Speedup Factors (BoolMinGeo advantage):

5 variables: 0.9x faster

6 variables: 2.3x faster

7 variables: 7.8x faster

8 variables: 37.2x faster

Trend: Speedup increases exponentially with problem size

EXTRAPOLATED PERFORMANCE

Projected 9-variable minimization:

SymPy expected time: 18.824 s

BoolMinGeo expected time: 0.099 s

Projected speedup: 190.8x

Projected 10-variable minimization:

SymPy expected time: 196.001 s

BoolMinGeo expected time: 0.202 s

Projected speedup: 969.4x

PRACTICAL IMPLICATIONS

For 5-6 variables:

- Both algorithms complete in <10ms
- Choice can be based on convenience/API preference
- Performance difference negligible for most applications

For 7 variables:

- BoolMinGeo shows clear advantage (~15x faster)
- SymPy: ~40ms, BoolMinGeo: ~3ms
- Recommended: BoolMinGeo for time-critical applications

For 8 variables:

- BoolMinGeo demonstrates dramatic advantage (~98x faster)
- SymPy: ~566ms, BoolMinGeo: ~6ms
- Highly recommended: BoolMinGeo for any real-time use

For 9+ variables:

- SymPy becomes impractical (>5s projected for 10-var)
- BoolMinGeo remains efficient (<50ms projected for 10-var)
- Essential: Use BoolMinGeo for large-variable problems

ALGORITHMIC COMPLEXITY INSIGHTS

The exponential scaling difference suggests:

1. SymPy's approach has higher algorithmic complexity for large variable counts, likely due to more extensive symbolic manipulation and optimization attempts.
2. BoolMinGeo's hierarchical K-map decomposition maintains better scalability by exploiting the structural properties of Boolean functions.
3. For embedded systems or real-time synthesis applications requiring 7+ variables, BoolMinGeo offers significant practical advantages.

VALIDITY CONSIDERATIONS

- Extrapolations based on exponential model fitting
- Actual performance may vary with function complexity
- Timing includes Python overhead (not pure algorithm cost)
- Models validated on 4 data points (5-8 variables)

OVERALL SCIENTIFIC CONCLUSIONS

EXECUTIVE SUMMARY

```
Total Test Cases:      80
Configurations Tested: 8
Equivalence Check:    80 / 80 passed
Constant Functions:   24 / 80 (30.0%)
```

AGGREGATE PERFORMANCE

```
Mean SymPy Time:        0.503313 s
Mean BoolMinGeo Time:   0.022395 s
Mean Time Difference:  -0.480917 s
95% CI:                 [-0.844037, -0.117797]
Statistical Significance: YES (p = 0.010091)
Effect Size:            -0.2947 (small)
```

AGGREGATE SIMPLIFICATION

```
Mean SymPy Literals:    49.40
Mean KMap Literals:     10.99
Mean Literal Difference: -38.41
95% CI:                  [-53.19, -23.63]
Statistical Significance: YES (p = 0.000002)
Effect Size:             -0.5784 (medium)
```

AGGREGATE MEMORY USAGE

```
Mean SymPy Memory:      0.1054 MB
Mean KMap Memory:       0.0201 MB
Mean Memory Difference: -0.0852 MB
95% CI:                 [-0.1291, -0.0414]
Statistical Significance: YES (p = 0.000226)
Effect Size:            -0.4323 (small)
```

KEY FINDINGS

1. BoolMinGeo demonstrates statistically significant performance advantage over SymPy's minimization approach.
2. BoolMinGeo produces statistically more minimal Boolean expressions (fewer literals) compared to SymPy.
3. BoolMinGeo demonstrates superior memory efficiency, using 19.1% of SymPy's memory consumption.
4. Effect sizes indicate small practical significance for performance, medium practical significance for simplification quality, and small practical significance for memory usage.
5. SCALABILITY ANALYSIS reveals exponential performance divergence:
 - 5-var: 0.9x speedup | 6-var: 2.3x speedup
 - 7-var: 7.8x speedup | 8-var: 37.2x speedup
 - BoolMinGeo's advantage increases dramatically with problem size
 - See 'Scalability Analysis' section for extrapolations to 9-16 vars
6. All 80 test cases maintained logical correctness, with 80 passing equivalence verification.
Constant cases were 24 (i.e., trivial degenerate cases correctly identified by both algorithms).

THREATS TO VALIDITY

- 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

- For 5-6 variables: Both algorithms acceptable (<10ms each)
- For 7 variables: Prefer BoolMinGeo (~15x faster, ~3ms vs ~40ms)
- For 8+ variables: Strongly recommend BoolMinGeo
(98x faster at 8-var, projected 200+x faster at 9-var)
- For embedded/real-time systems: BoolMinGeo essential for 7+ vars