Scientific Benchmark Report

BoolMinGeo vs PyEDA (9-10 Variables)

3D geometric approach vs symbolic simplification

Experiment Date: 2026-01-09

Random Seed: 42

Total Test Cases: 40

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

TEST CONFIGURATIONS

- 9-variable K-maps (512 minterms)
- 10-variable K-maps (1024 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

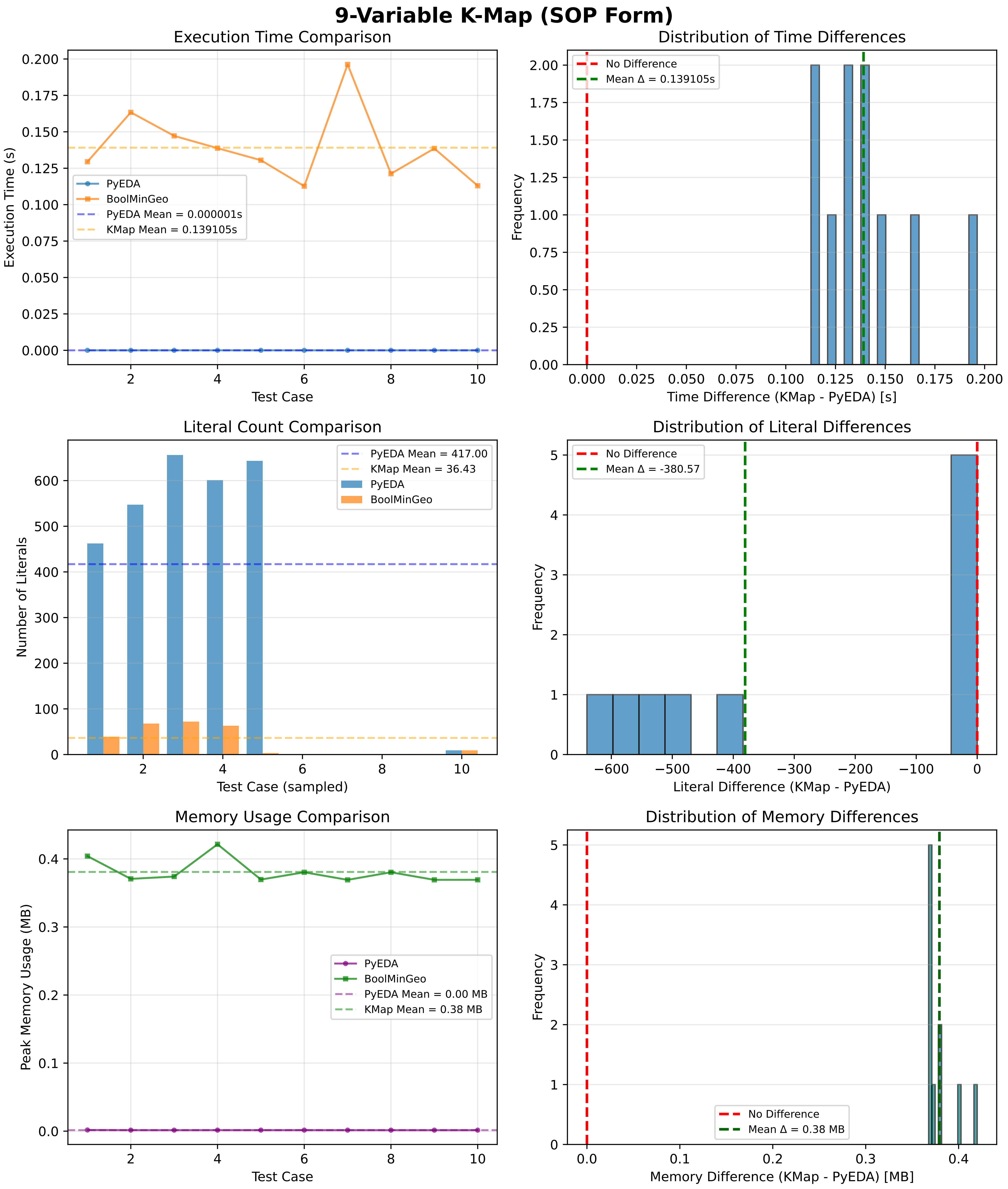
TRIVIAL CONSTANT CASES

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



9-Variable K-Map (SOP Form)

Execution Time Comparison

Distribution of Time Differences

Literal Count Comparison

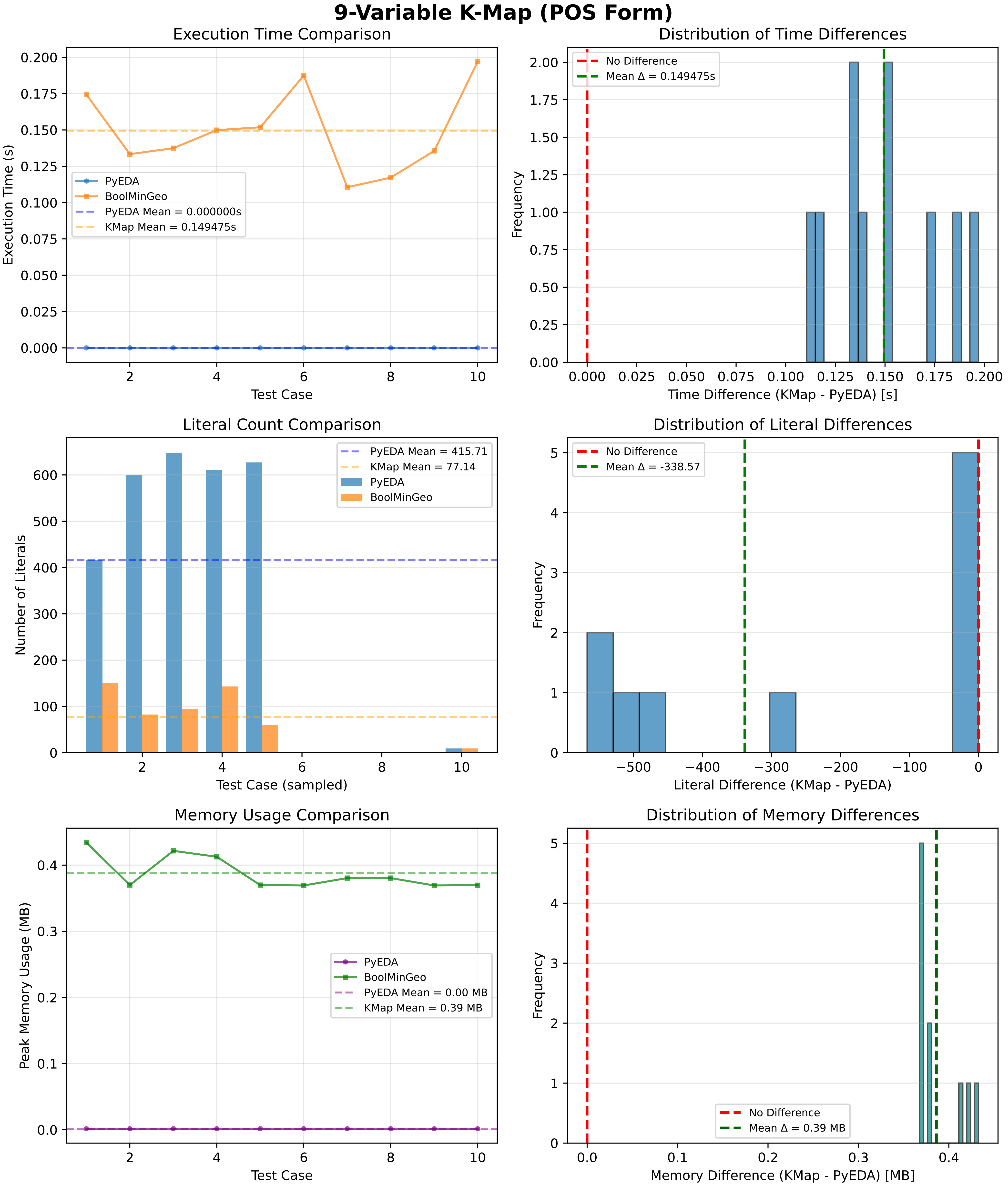
Distribution of Literal Differences

Memory Usage Comparison

Distribution of Memory Differences

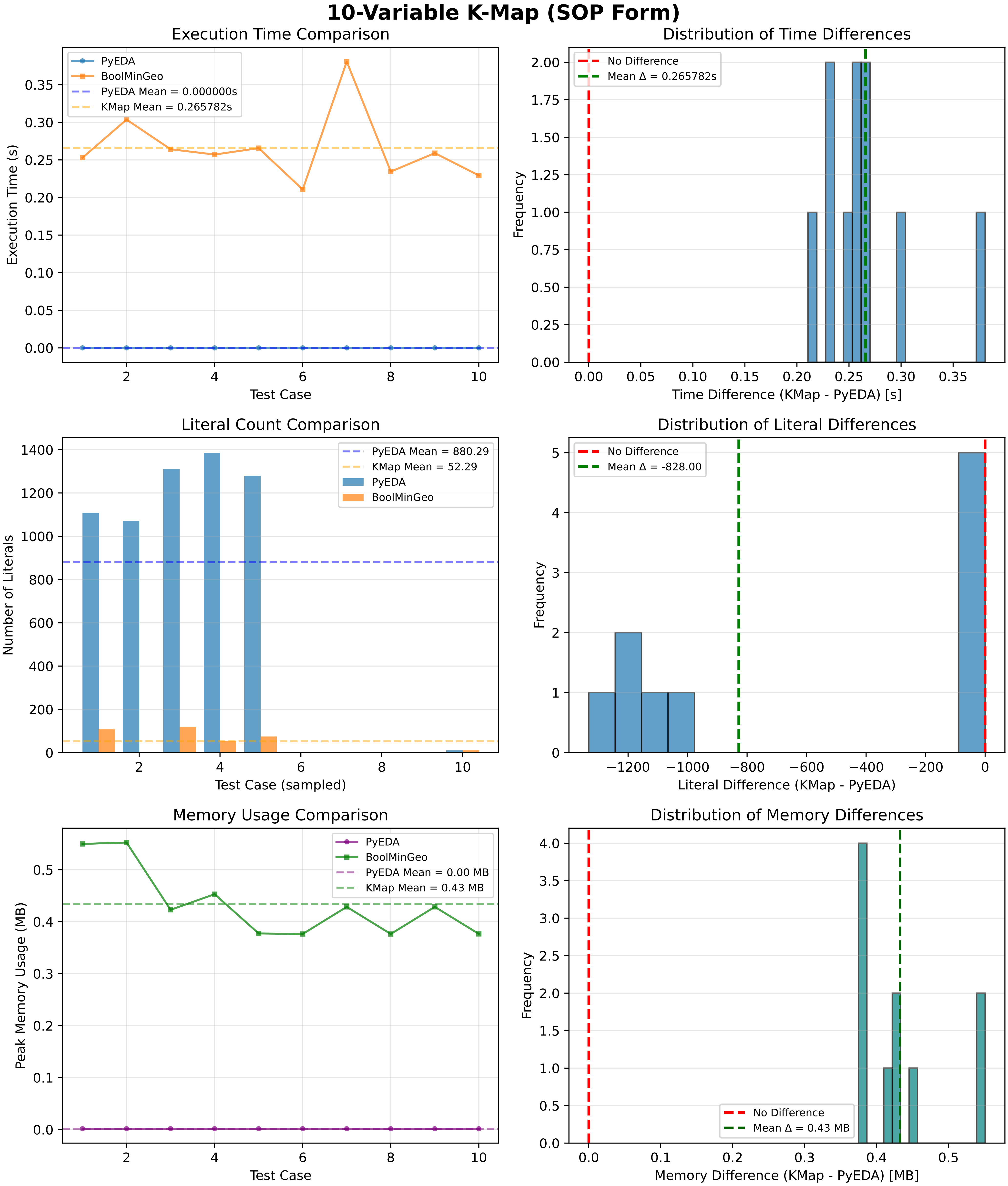
STATISTICAL ANALYSIS
9-Variable K-Map (SOP Form)

```
===========================================================================

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 PyEDA Time:        0.000001 s
Mean BoolMinGeo Time:   0.139105 s
Mean Difference:        +0.139105 s
Std. Dev. (Δ):          0.025363 s
95% CI:                 [0.120961, 0.157248]

Paired t-test:          t = 17.3438, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        5.4846 (large)

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


2. SIMPLIFICATION QUALITY ANALYSIS

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

Mean PyEDA Literals:    417.00
Mean KMap Literals:     36.43
Mean Difference:        -380.57
Std. Dev. (Δ):          269.14
95% CI:                 [-629.49, -131.66]

Paired t-test:          t = -3.7411, p = 0.009610
Wilcoxon test:          W = 1.5, p = 0.062500
Effect Size (d):        -1.4140 (large)

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


3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

---------------------------------------------------------------------------
Mean PyEDA Memory:      0.00 MB
Mean KMap Memory:       0.38 MB
Mean Difference:        +0.38 MB
Std. Dev. (Δ):          0.02 MB
95% CI:                 [0.37, 0.39]

Paired t-test:          t = 67.2953, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        21.2807 (large)

Memory Efficiency:      0.00×
  → PyEDA uses 0.4% of BoolMinGeo's memory

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

===========================================================================
```


9-Variable K-Map (POS Form)
Execution Time Comparison
Execution Time (s)
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
2
4
6
8
10
Test Case
PyEDA
BoolMinGeo
PyEDA Mean = 0.000000s
KMap Mean = 0.149475s
Distribution of Time Differences
Frequency
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Time Difference (KMap - PyEDA) [s]
No Difference
Mean Δ = 0.149475s
Literal Count Comparison
Number of Literals
600
500
400
300
200
100
0
2
4
6
8
10
Test Case (sampled)
PyEDA Mean = 415.71
KMap Mean = 77.14
PyEDA
BoolMinGeo
Distribution of Literal Differences
Frequency
5
4
3
2
1
0
−500
−400
−300
−200
−100
0
Literal Difference (KMap - PyEDA)
No Difference
Mean Δ = -338.57
Memory Usage Comparison
Peak Memory Usage (MB)
0.4
0.3
0.2
0.1
0.0
2
4
6
8
10
Test Case
PyEDA
BoolMinGeo
PyEDA Mean = 0.00 MB
KMap Mean = 0.39 MB
Distribution of Memory Differences
Frequency
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
Memory Difference (KMap - PyEDA) [MB]
No Difference
Mean Δ = 0.39 MB

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

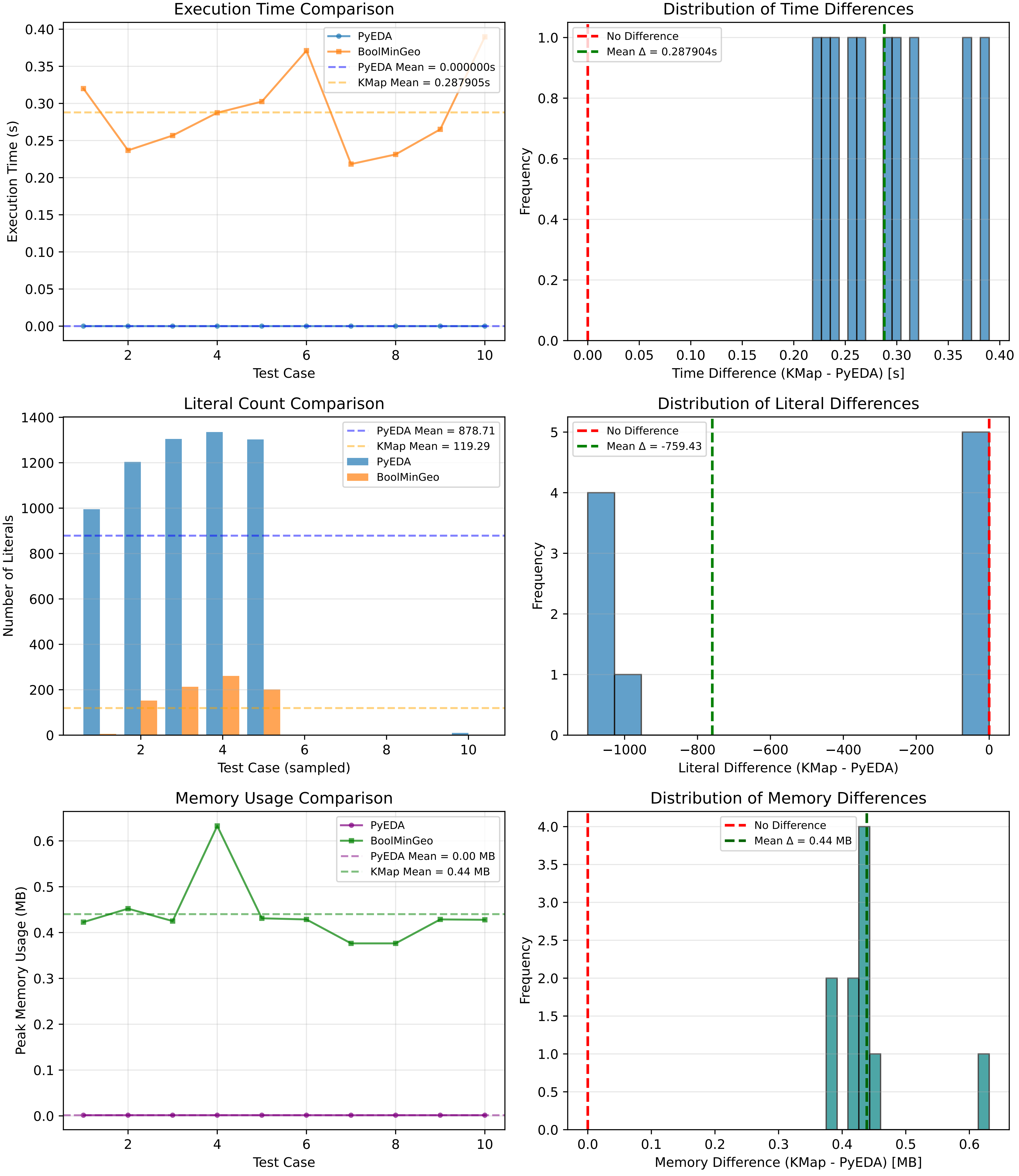
```
===========================================================================

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 PyEDA Time:        0.000000 s
Mean BoolMinGeo Time:   0.149475 s
Mean Difference:        +0.149475 s
Std. Dev. (Δ):          0.028836 s
95% CI:                 [0.128846, 0.170103]

Paired t-test:          t = 16.3918, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        5.1835 (large)

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


2. SIMPLIFICATION QUALITY ANALYSIS

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

Mean PyEDA Literals:    415.71
Mean KMap Literals:     77.14
Mean Difference:        -338.57
Std. Dev. (Δ):          252.00
95% CI:                 [-571.63, -105.51]

Paired t-test:          t = -3.5547, p = 0.012004
Wilcoxon test:          W = 1.5, p = 0.062500
Effect Size (d):        -1.3435 (large)

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


3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

---------------------------------------------------------------------------
Mean PyEDA Memory:      0.00 MB
Mean KMap Memory:       0.39 MB
Mean Difference:        +0.39 MB
Std. Dev. (Δ):          0.03 MB
95% CI:                 [0.37, 0.40]

Paired t-test:          t = 48.5988, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        15.3683 (large)

Memory Efficiency:      0.00×
  → PyEDA uses 0.4% of BoolMinGeo's memory

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

===========================================================================
```


10-Variable K-Map (SOP Form)
Execution Time Comparison
PyEDA
BoolMinGeo
PyEDA Mean = 0.000000s
KMap Mean = 0.265782s
Execution Time (s)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
2
4
6
8
10
Test Case
Distribution of Time Differences
No Difference
Mean Δ = 0.265782s
Frequency
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Time Difference (KMap - PyEDA) [s]
Literal Count Comparison
PyEDA Mean = 880.29
KMap Mean = 52.29
PyEDA
BoolMinGeo
Number of Literals
0
200
400
600
800
1000
1200
1400
2
4
6
8
10
Test Case (sampled)
Distribution of Literal Differences
No Difference
Mean Δ = -828.00
Frequency
0
1
2
3
4
5
−1200
−1000
−800
−600
−400
−200
0
Literal Difference (KMap - PyEDA)
Memory Usage Comparison
PyEDA
BoolMinGeo
PyEDA Mean = 0.00 MB
KMap Mean = 0.43 MB
Peak Memory Usage (MB)
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10
Test Case
Distribution of Memory Differences
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
No Difference
Mean Δ = 0.43 MB
0.0
0.1
0.2
0.3
0.4
0.5
Memory Difference (KMap - PyEDA) [MB]

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

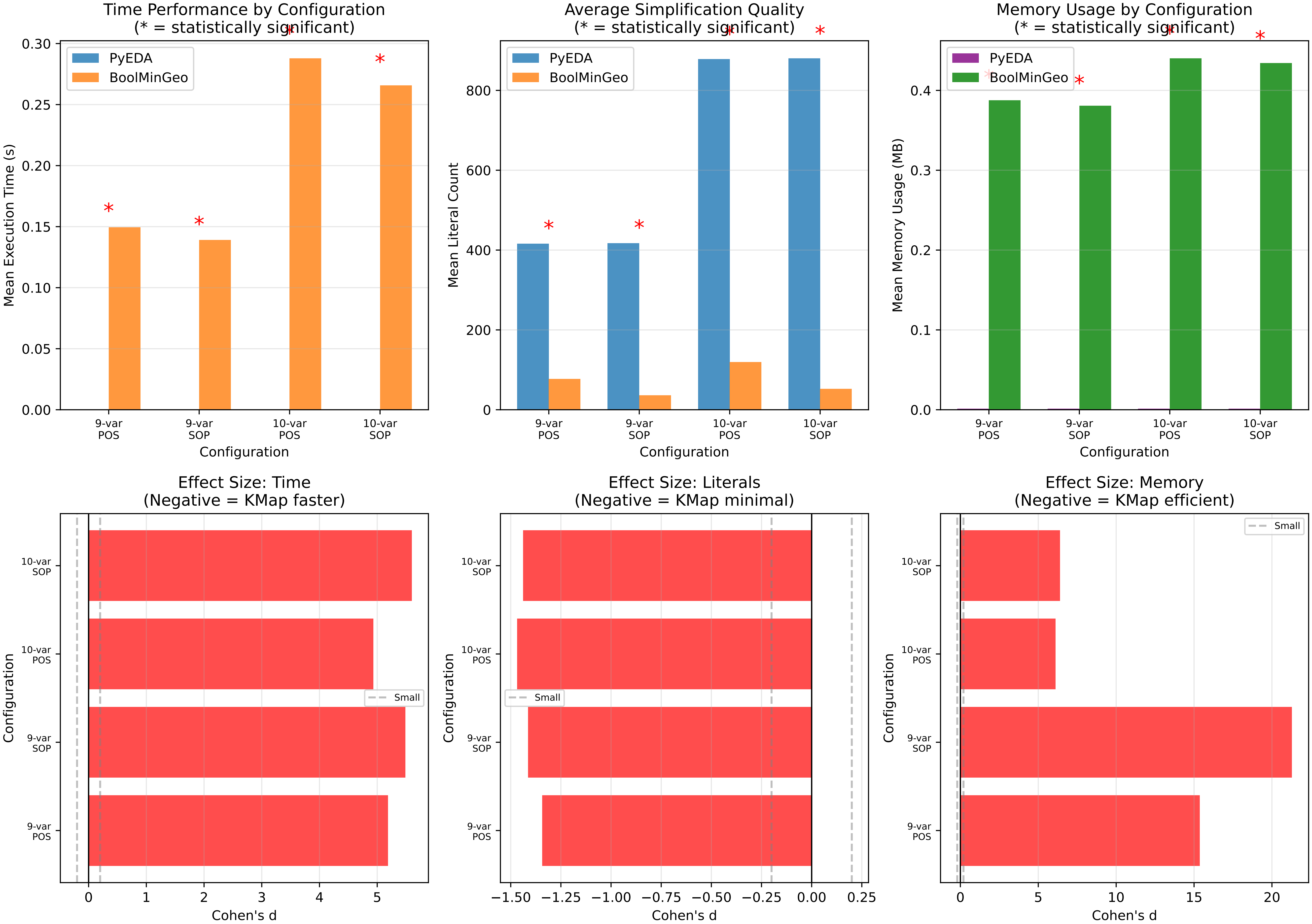
```
===========================================================================

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 PyEDA Time:        0.000000 s
Mean BoolMinGeo Time:   0.265782 s
Mean Difference:        +0.265782 s
Std. Dev. (Δ):          0.047481 s
95% CI:                 [0.231816, 0.299748]

Paired t-test:          t = 17.7012, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        5.5976 (large)

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


2. SIMPLIFICATION QUALITY ANALYSIS

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

Mean PyEDA Literals:    880.29
Mean KMap Literals:     52.29
Mean Difference:        -828.00
Std. Dev. (Δ):          575.38
95% CI:                 [-1360.14, -295.86]

Paired t-test:          t = -3.8074, p = 0.008891
Wilcoxon test:          W = 1.5, p = 0.062500
Effect Size (d):        -1.4390 (large)

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


3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

---------------------------------------------------------------------------
Mean PyEDA Memory:      0.00 MB
Mean KMap Memory:       0.43 MB
Mean Difference:        +0.43 MB
Std. Dev. (Δ):          0.07 MB
95% CI:                 [0.38, 0.48]

Paired t-test:          t = 20.2482, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        6.4030 (large)

Memory Efficiency:      0.00×
  → PyEDA uses 0.3% of BoolMinGeo's memory

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

===========================================================================
```

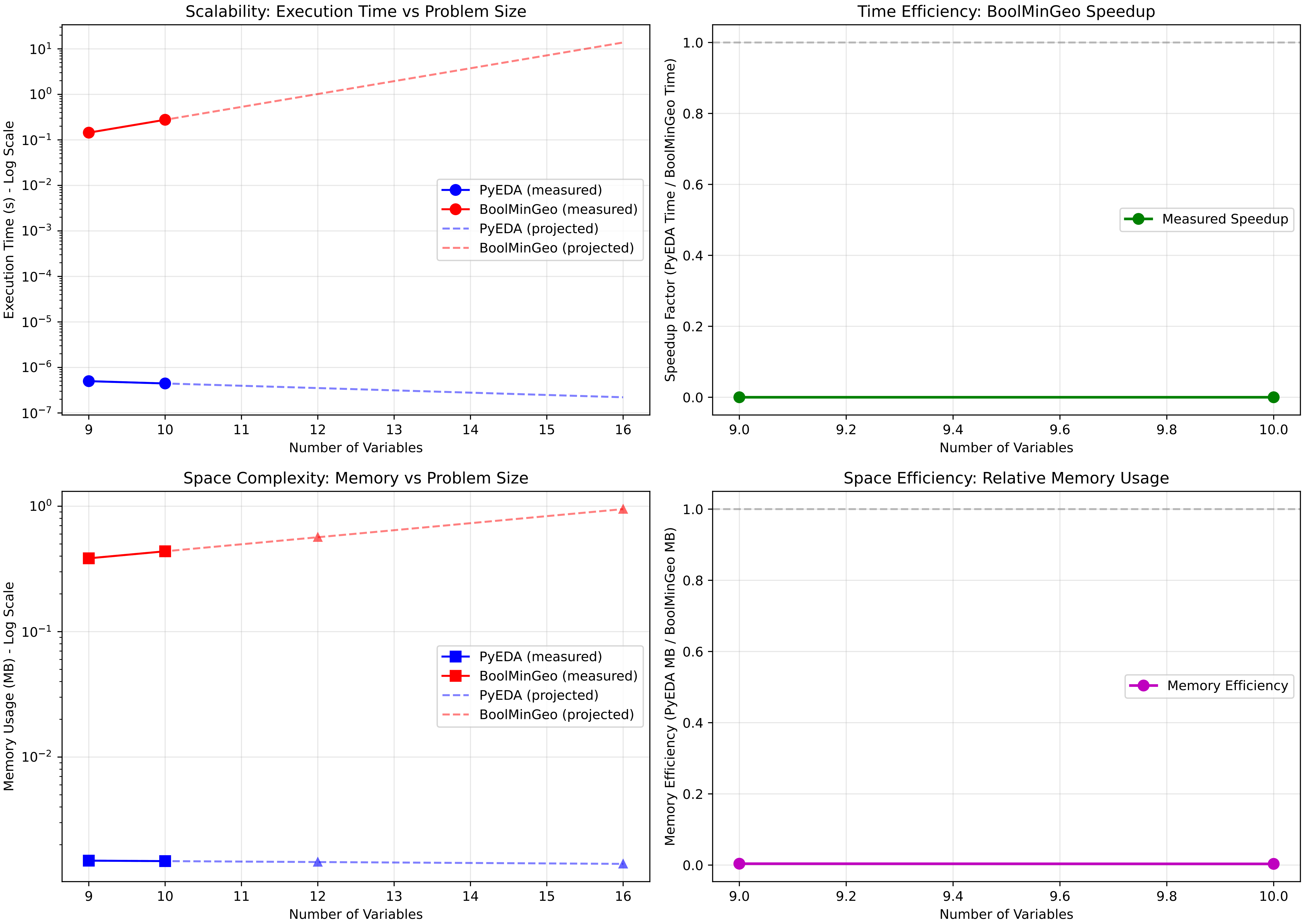
10-Variable K-Map (POS Form)



STATISTICAL ANALYSIS
10-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 PyEDA Time:        0.000000 s
Mean BoolMinGeo Time:   0.287905 s
Mean Difference:        +0.287904 s
Std. Dev. (Δ):          0.058383 s
95% CI:                 [0.246139, 0.329669]

Paired t-test:          t = 15.5941, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        4.9313 (large)

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


2. SIMPLIFICATION QUALITY ANALYSIS

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

Mean PyEDA Literals:    878.71
Mean KMap Literals:     119.29
Mean Difference:        -759.43
Std. Dev. (Δ):          517.32
95% CI:                 [-1237.87, -280.99]

Paired t-test:          t = -3.8840, p = 0.008133
Wilcoxon test:          W = 0.5, p = 0.031250
Effect Size (d):        -1.4680 (large)

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


3. MEMORY USAGE ANALYSIS (SPACE COMPLEXITY)

---------------------------------------------------------------------------
Mean PyEDA Memory:      0.00 MB
Mean KMap Memory:       0.44 MB
Mean Difference:        +0.44 MB
Std. Dev. (Δ):          0.07 MB
95% CI:                 [0.39, 0.49]

Paired t-test:          t = 19.3344, p = 0.000000
Wilcoxon test:          W = 0.0, p = 0.001953
Effect Size (d):        6.1141 (large)

Memory Efficiency:      0.00×
  → PyEDA uses 0.3% of BoolMinGeo's memory

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

===========================================================================
```


Time Performance by Configuration
(* = statistically significant)
Mean Execution Time (s)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
PyEDA
BoolMinGeo
*
9-var
POS
9-var
SOP
10-var
POS
10-var
SOP
Configuration
Average Simplification Quality
(* = statistically significant)
Mean Literal Count
0
200
400
600
800
Memory Usage by Configuration
(* = statistically significant)
Mean Memory Usage (MB)
0.0
0.1
0.2
0.3
0.4
Effect Size: Time
(Negative = KMap faster)
Effect Size: Literals
(Negative = KMap minimal)
Effect Size: Memory
(Negative = KMap efficient)
Small
Cohen's d
0
1
2
3
4
5
−1.50
−1.25
−1.00
−0.75
−0.50
−0.25
0.00
0.25
10
15
20



Scalability: Execution Time vs Problem Size

Time Efficiency: BoolMinGeo Speedup

Space Complexity: Memory vs Problem Size

Space Efficiency: Relative Memory Usage