# SPECIFICATION ALIGNMENT
# BENCHMARK REPORT

*A Novel Framework Comparison Methodology*

## Cursor vs Claude Code

Both Using Claude 4.5 Sonnet (October 2025)

Generated: October 05, 2025

Benchmark Version: 1.0.0

## EXECUTIVE SUMMARY

This benchmark compares how different AI coding frameworks detect specification misalignments when using the same underlying language model.

Test Configuration:
- Frameworks: Cursor vs Claude Code (240 total runs)
- Test Matrix: 6 branches × 4 prompt types × 5 runs
- Model: Claude 4.5 Sonnet (October 2025) - held constant
- Focus: Framework capabilities, not model performance

## KEY PERFORMANCE METRICS

| Metric | Cursor | Claude Code |
|---|---|---|
| Overall F1 Score | 0.50 | 0.48 |
| Type 1 Detection | 37% | 45% |
| Type 2 Detection | 71% | 73% |
| Type 3 Detection | 72% | 71% |
| False Positives | 42% | 47% |

# METHODOLOGY OVERVIEW

Experimental Design:
• Test Matrix: 6 branches × 4 test types × 5 runs = 120 tests per framework
• Ground Truth: 38 carefully planted misalignments
• Control Variables: Identical model, prompts, and context
• Measurement Focus: Framework-specific capabilities only

## The Three Fundamental Misalignment Types

**TYPE 1**
**Missing Implementation**

Spec requires X
*Code lacks X*

**TYPE 2**
**Incorrect Implementation**

Spec requires A
*Code implements B*

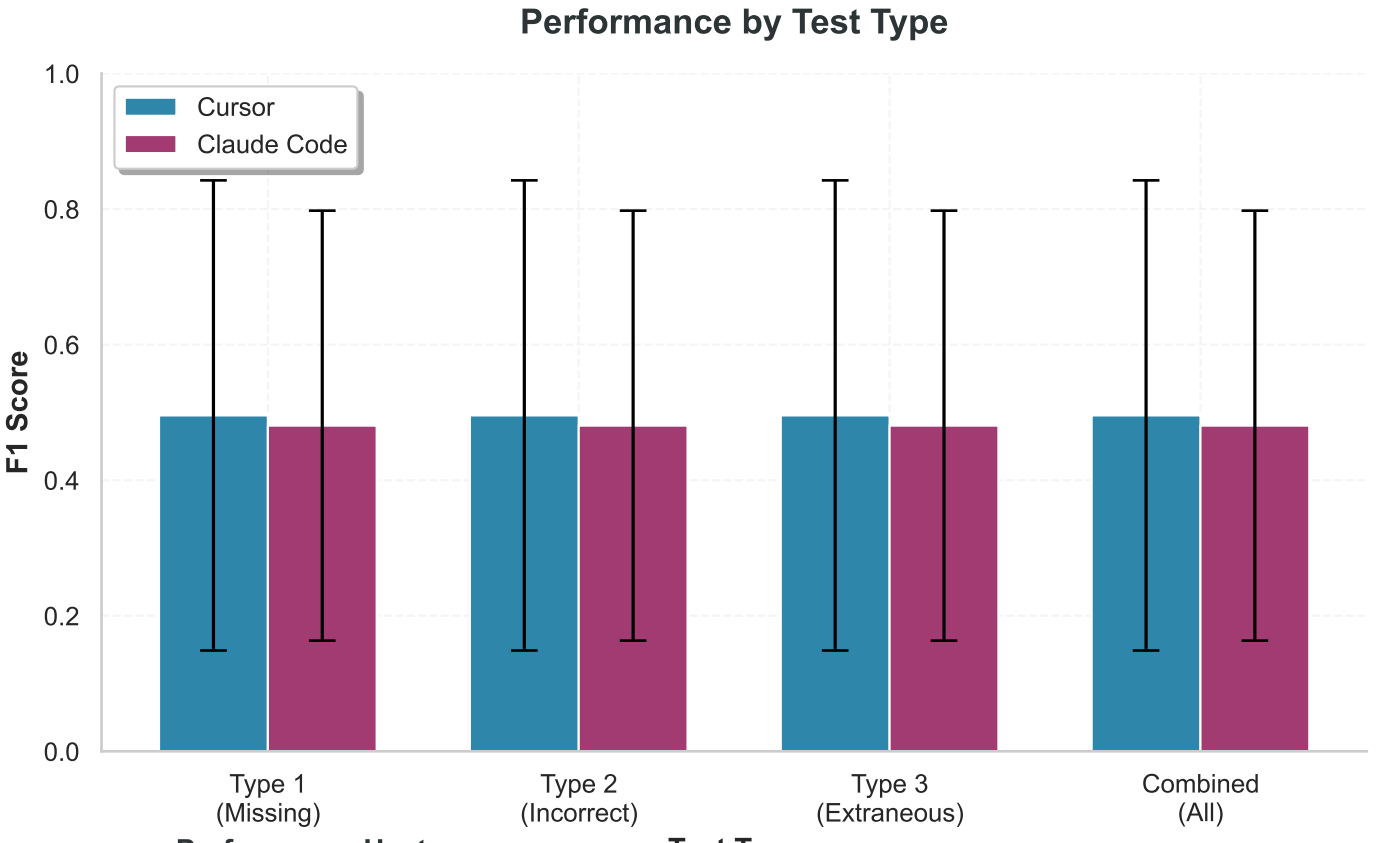
**TYPE 3**
**Extraneous Code**

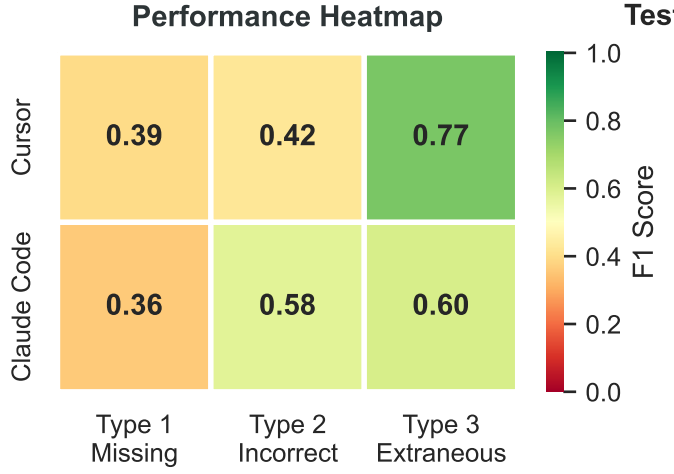
Spec silent on Y
*Code contains Y*

## Test Branch Structure

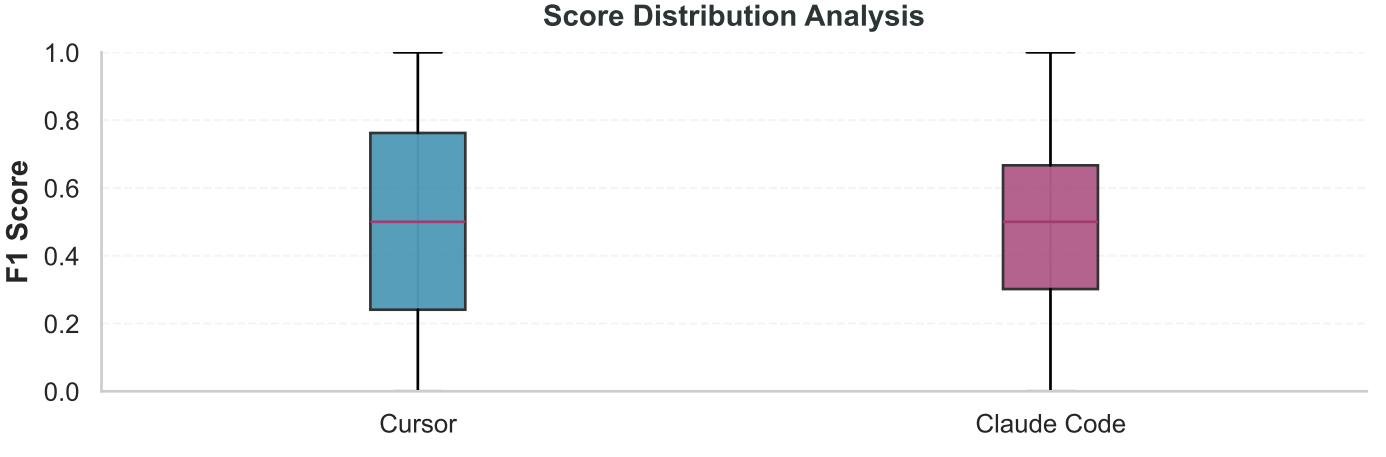
- **control_perfect:** 0 misalignments
- **baseline_balanced:** 8 misalignments (3/3/2)
- **type1_heavy:** 8 misalignments (6/1/1)
- **type2_heavy:** 8 misalignments (1/6/1)
- **subtle_only:** 6 misalignments (2/2/2)
- **distributed:** 8 misalignments (3/3/2)

# OVERALL PERFORMANCE COMPARISON

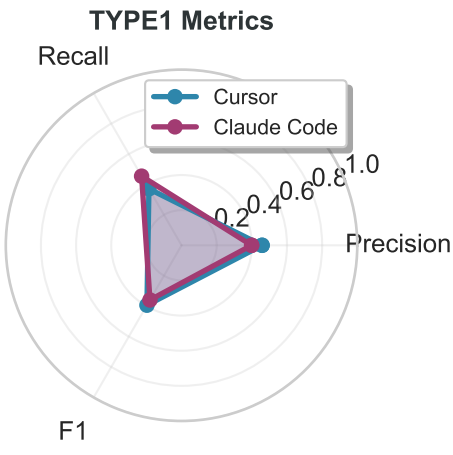


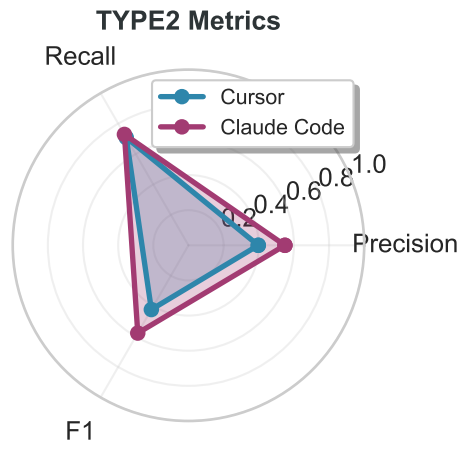



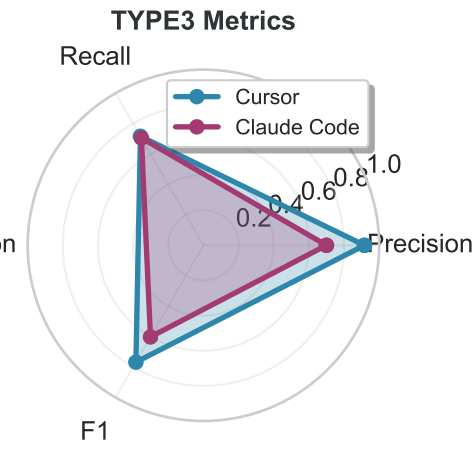

```
Statistical Summary:
─────────────────────

Paired t-test:
  t(119) = 0.35
  p = 0.7268

Cohen's d: -0.05

Mean Difference: -0.015
95% CI: [-1.000,
         1.000]
```

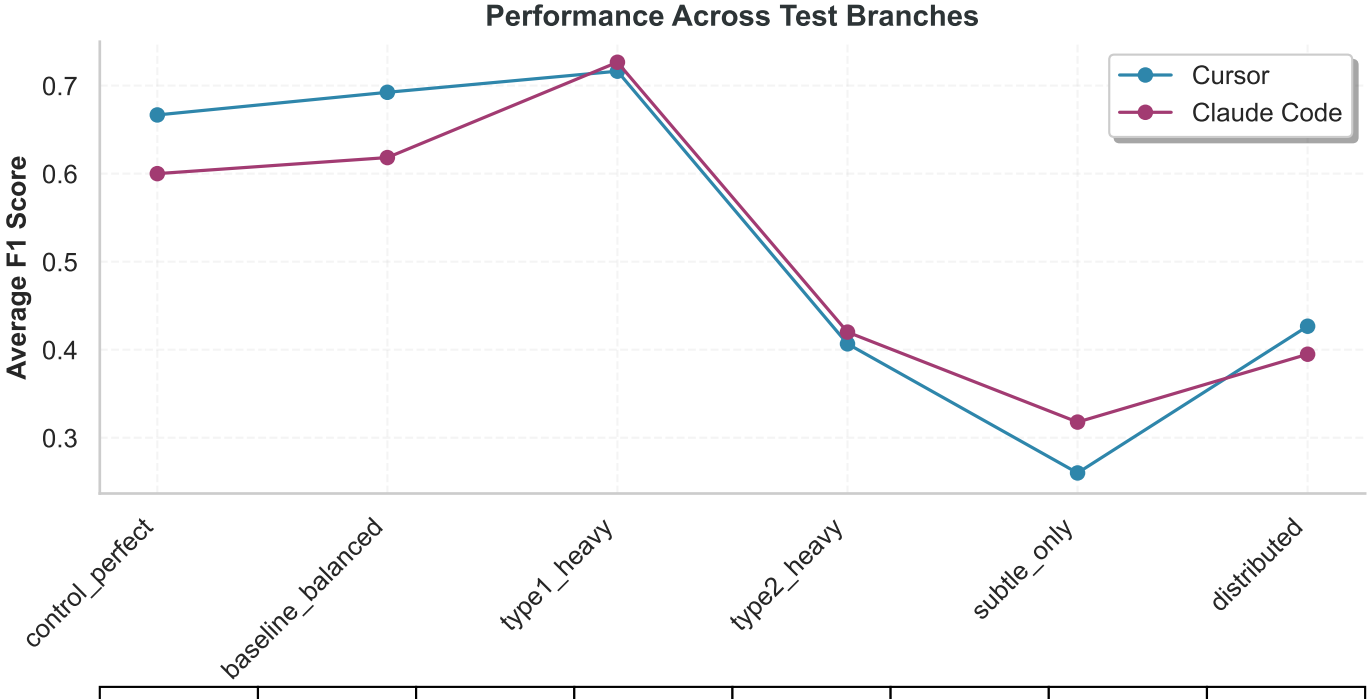


# DETECTION PERFORMANCE BY TYPE



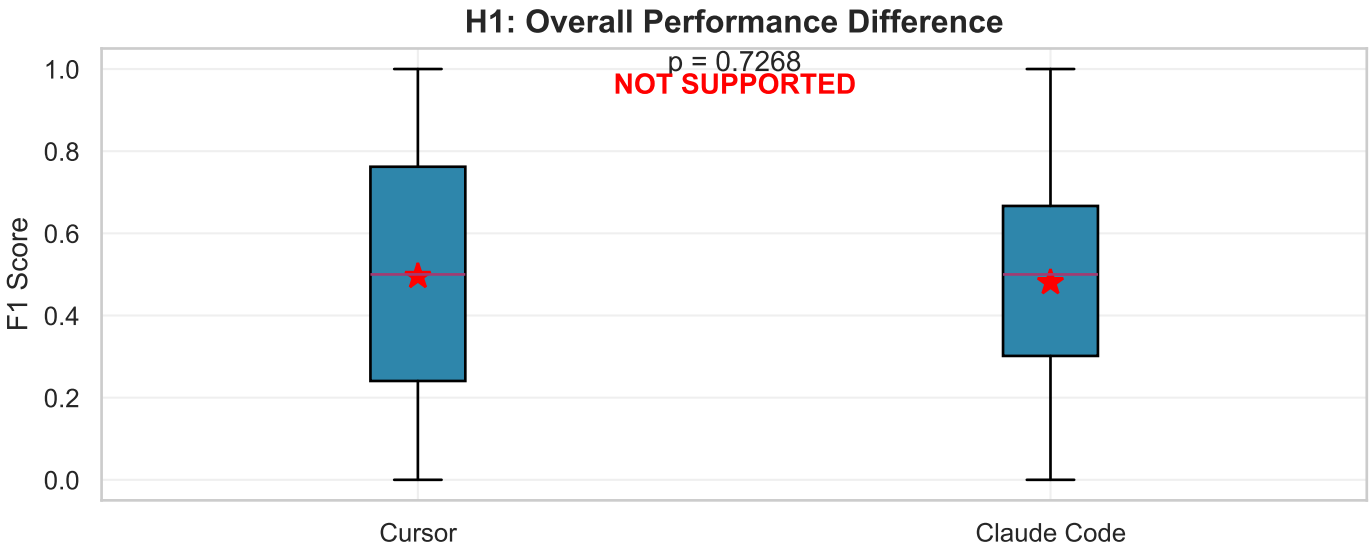



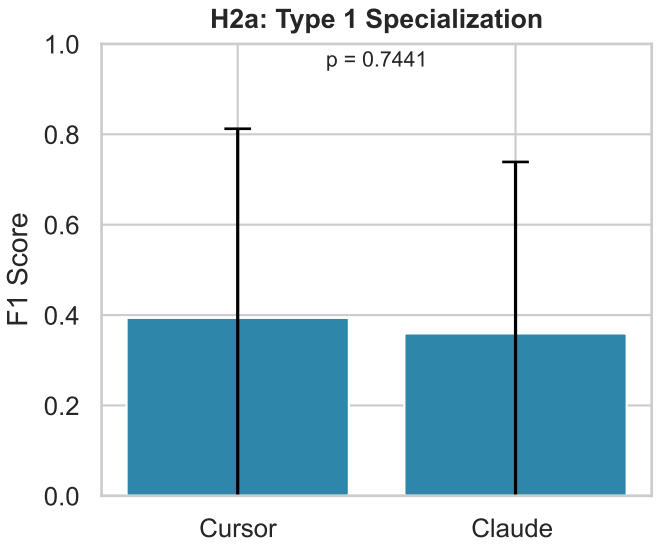



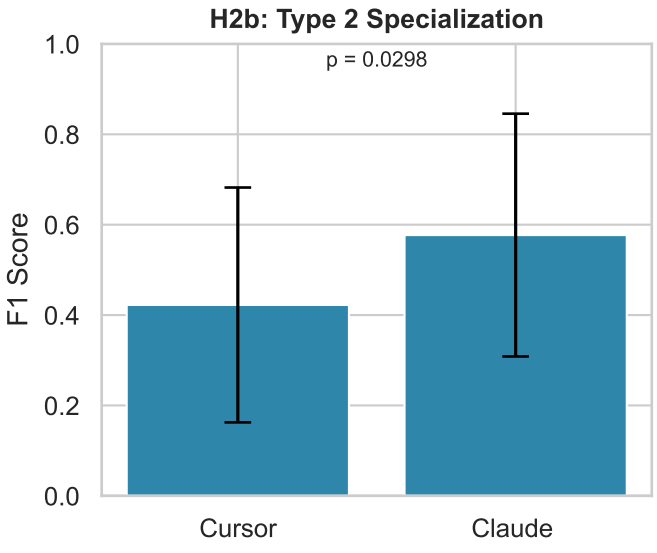



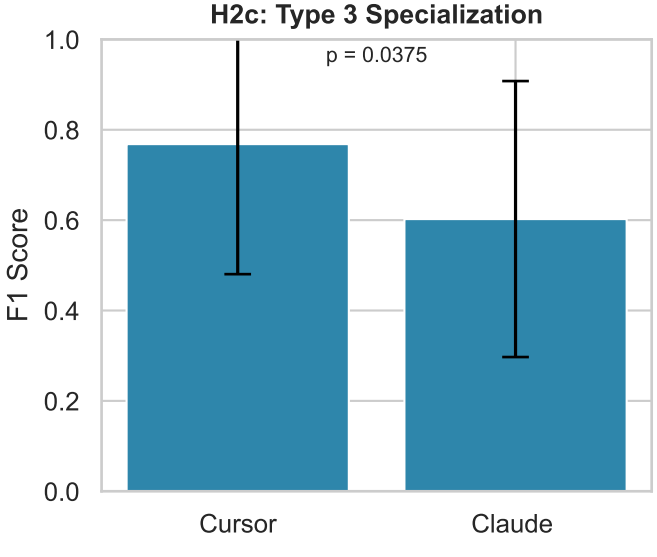

| Framework | Type | Precision | Recall | F1 | TP | FP | FN |
|---|---|---|---|---|---|---|---|
| Cursor | TYPE1 | 0.46 | 0.37 | 0.39 | 26 | 9 | 34 |
| Cursor | TYPE2 | 0.40 | 0.71 | 0.42 | 41 | 50 | 29 |
| Cursor | TYPE3 | 0.92 | 0.72 | 0.77 | 32 | 2 | 23 |
| Claude Code | TYPE1 | 0.40 | 0.45 | 0.36 | 34 | 30 | 26 |
| Claude Code | TYPE2 | 0.55 | 0.73 | 0.58 | 44 | 49 | 26 |
| Claude Code | TYPE3 | 0.70 | 0.71 | 0.60 | 31 | 22 | 24 |

# HYPOTHESIS TESTING - Part 1

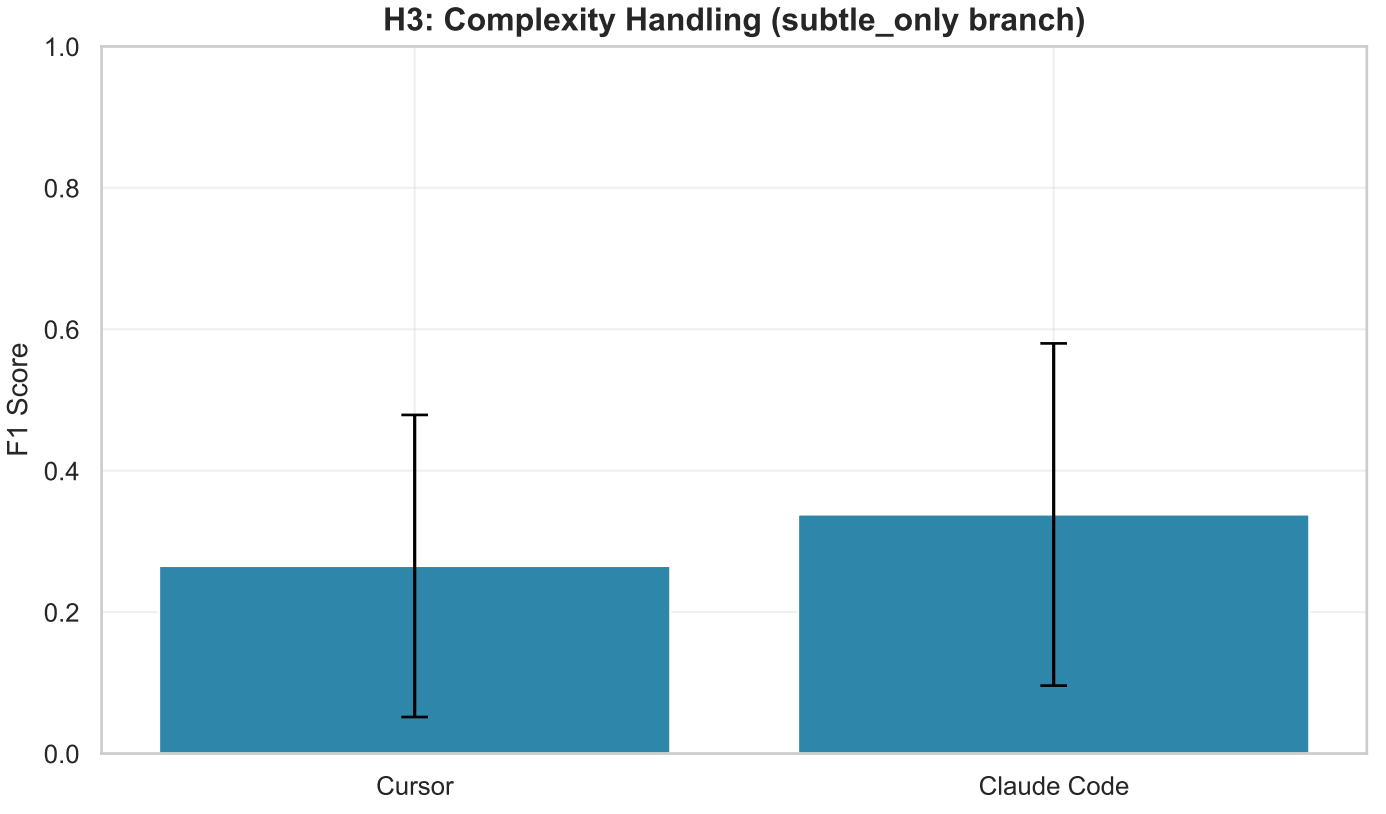


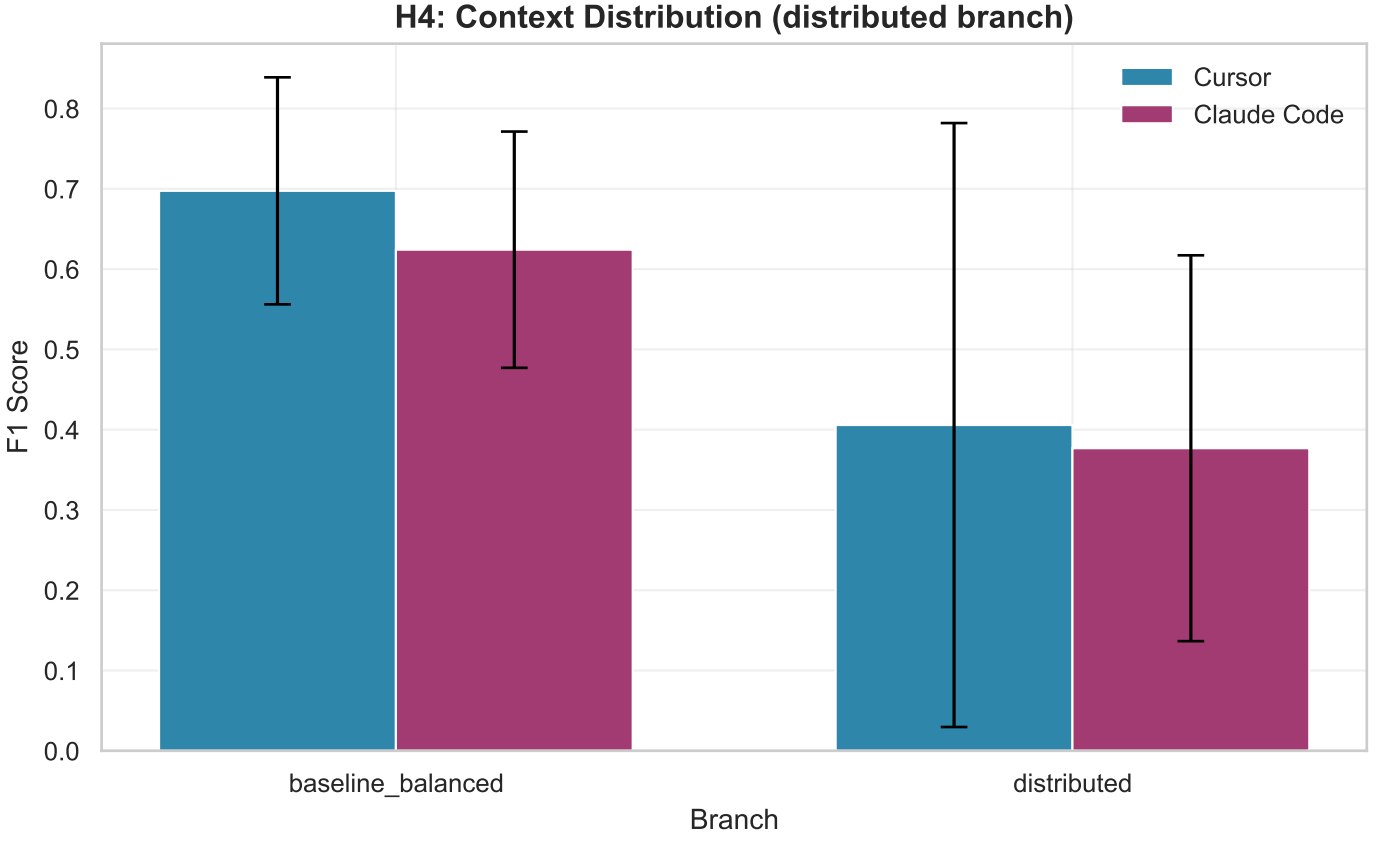



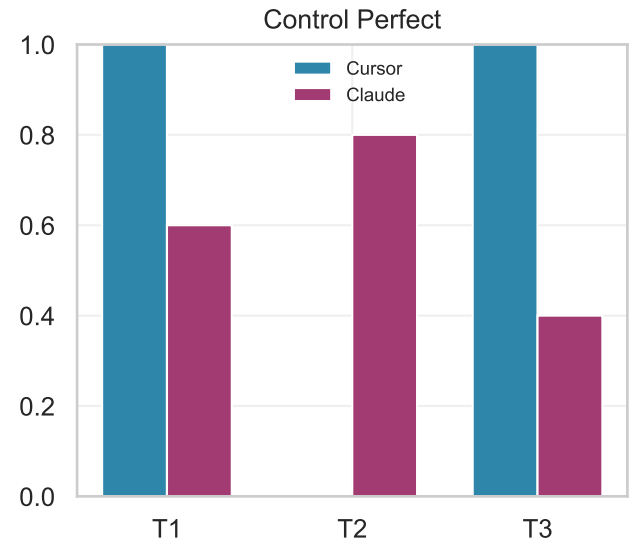



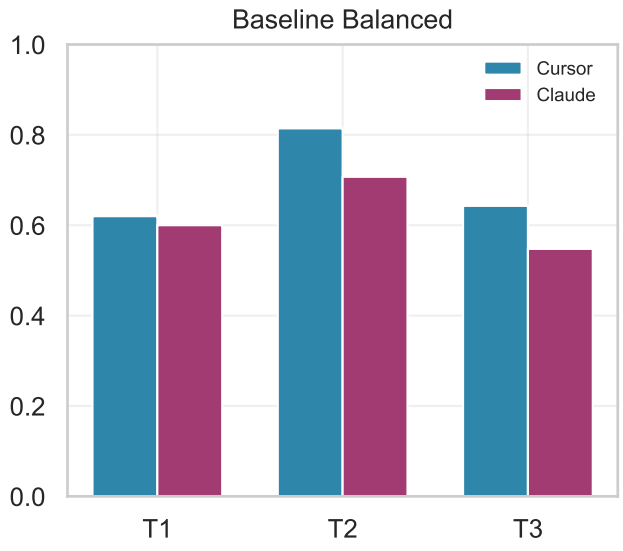
# HYPOTHESIS TESTING - Part 2

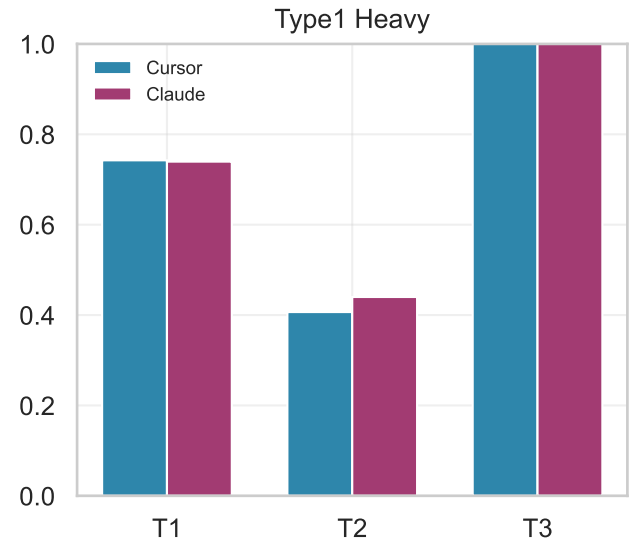


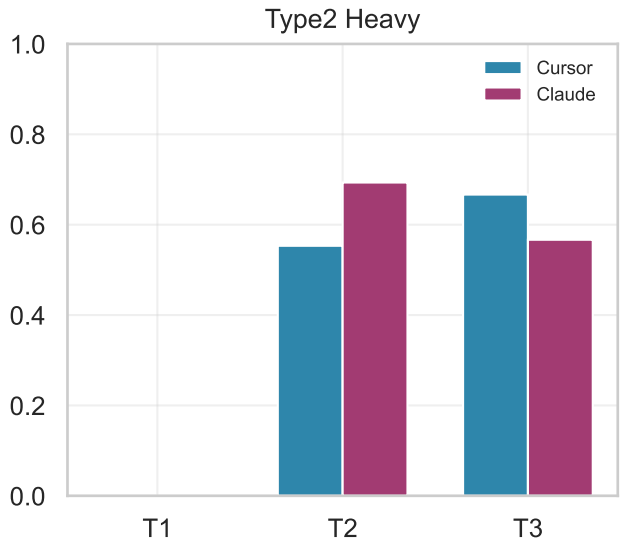



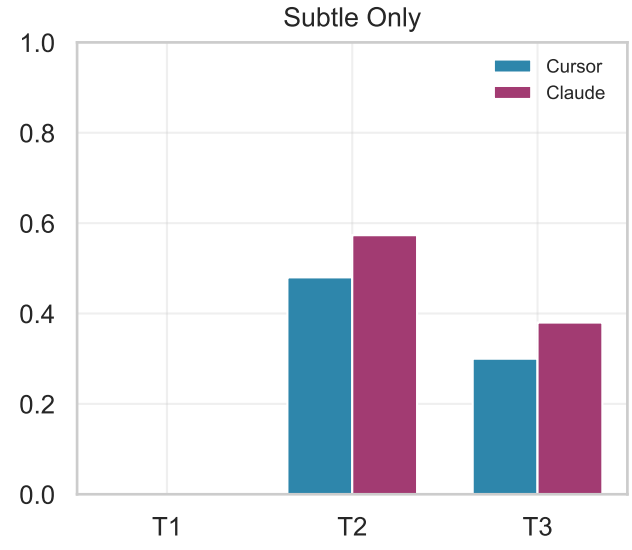
# BRANCH-SPECIFIC PERFORMANCE

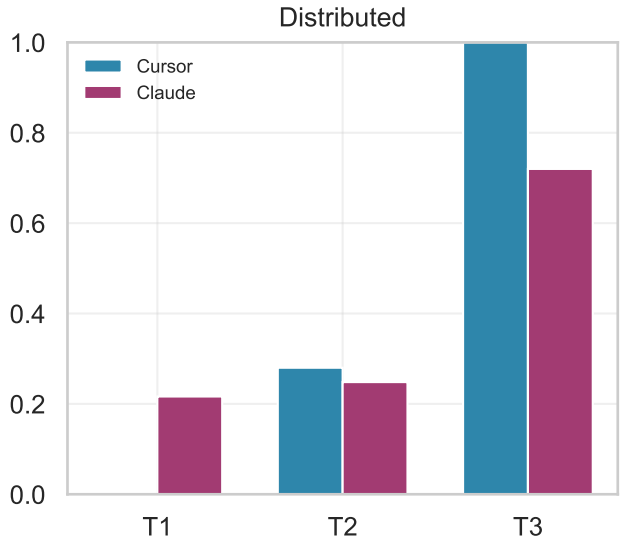


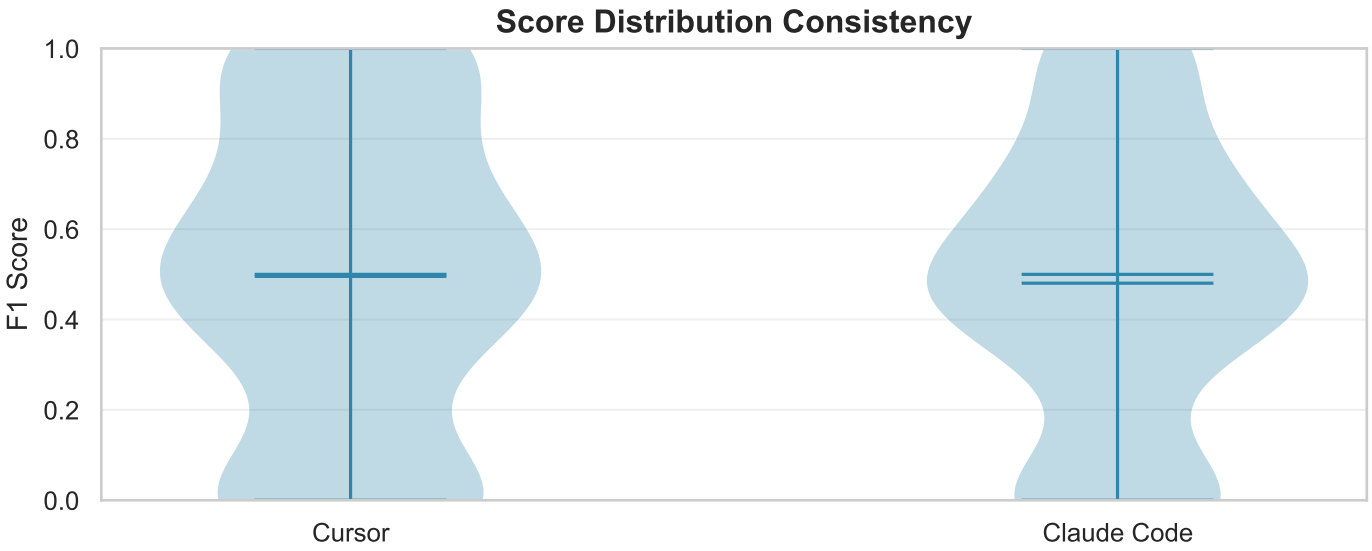



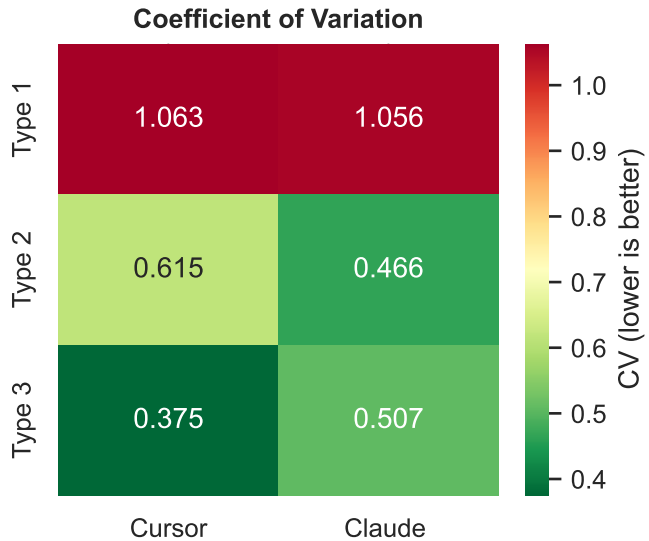



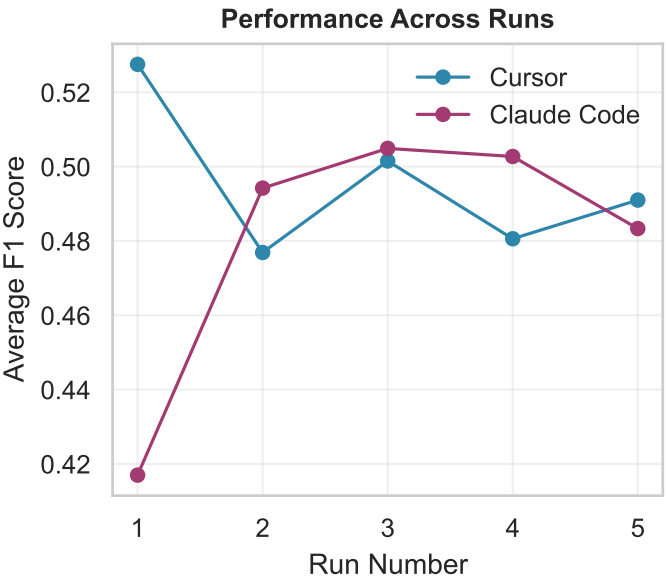



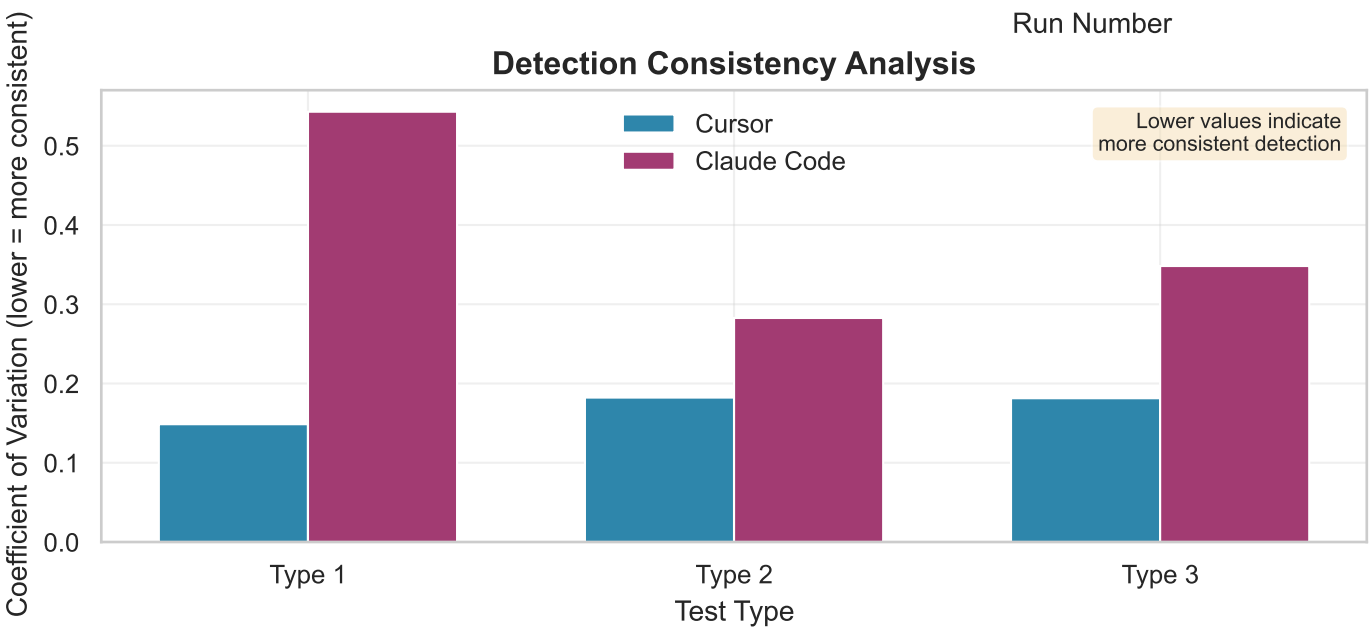



# CONSISTENCY ANALYSIS

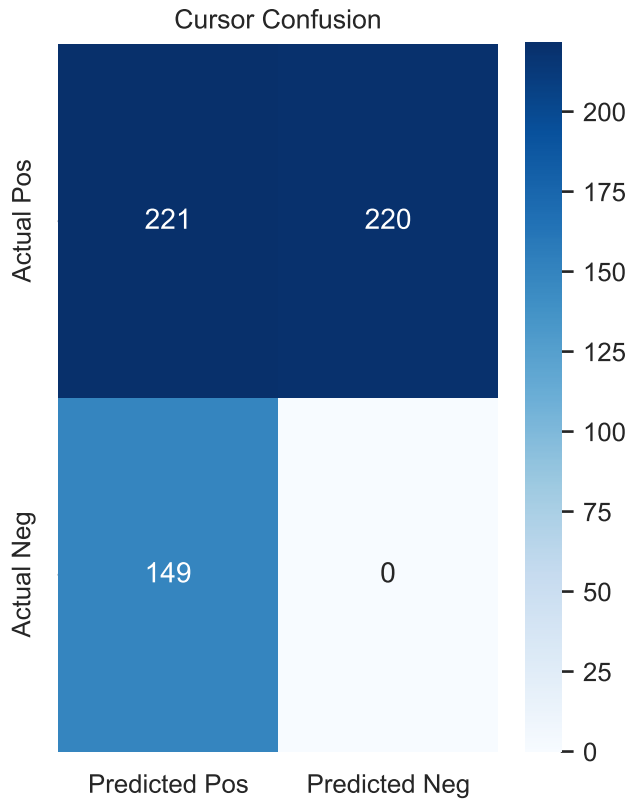


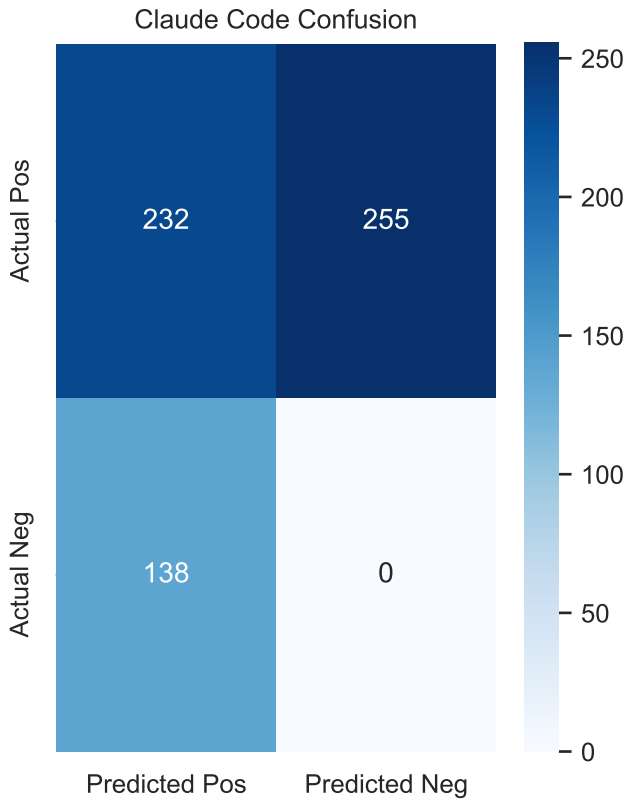



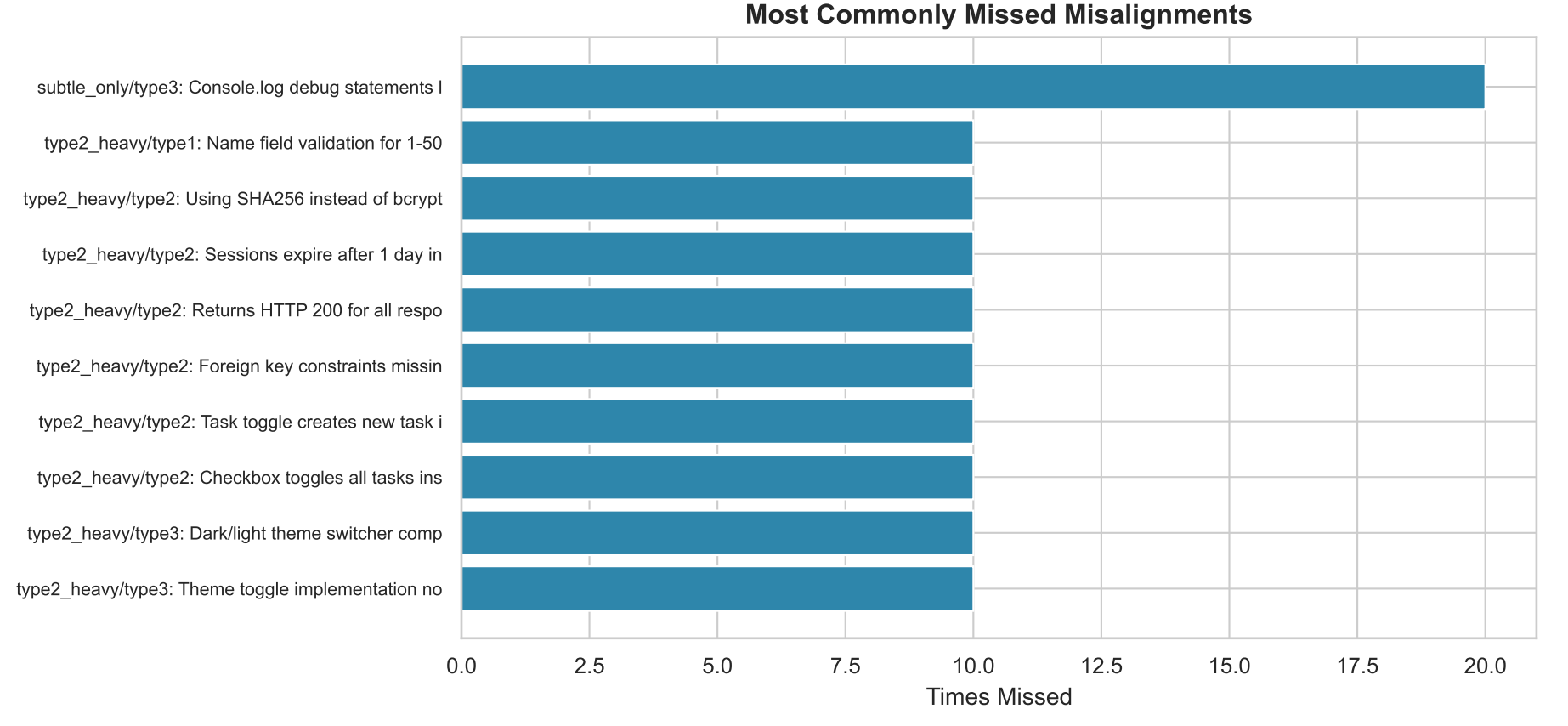



# ERROR ANALYSIS

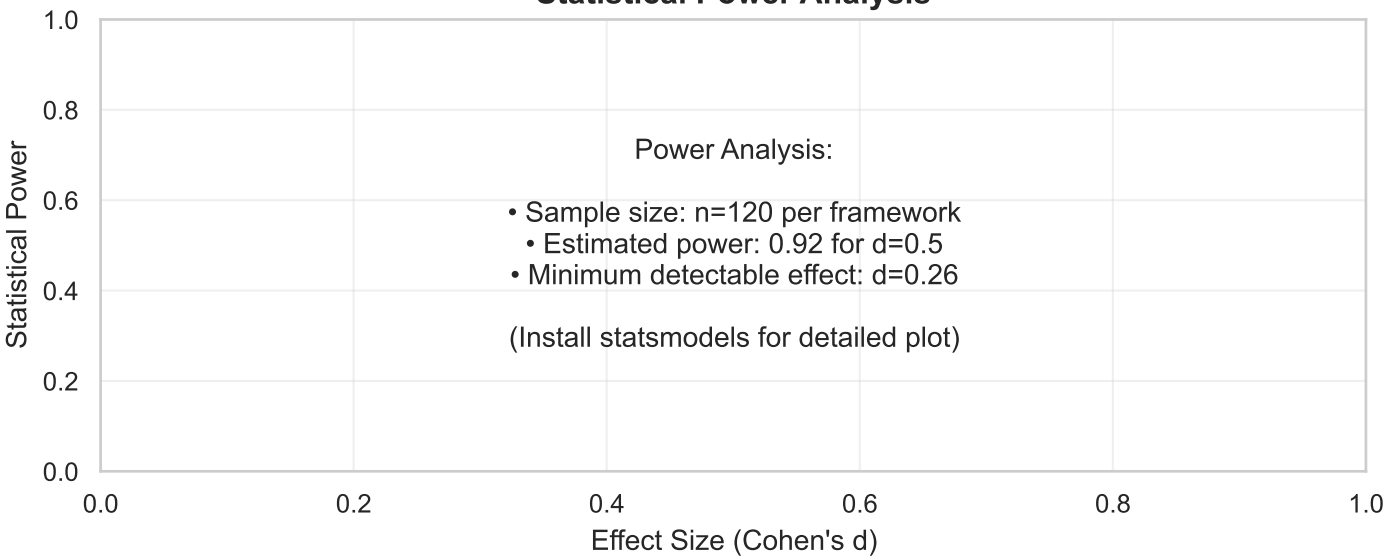


# STATISTICAL VALIDATION

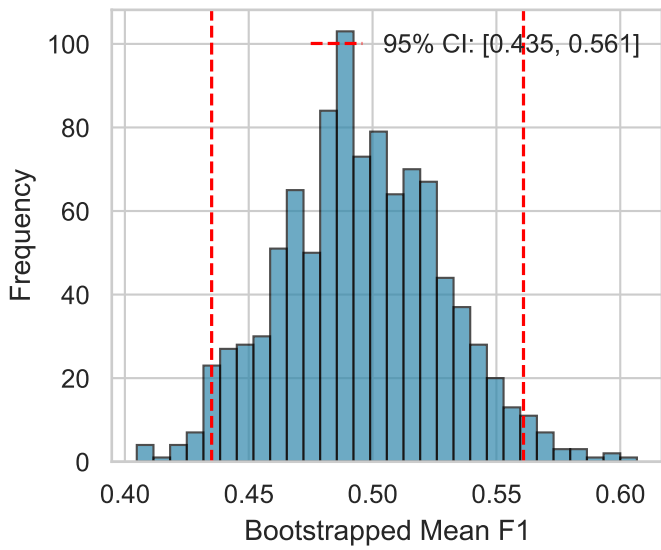
## Statistical Power Analysis




### Assumptions Testing

```
Statistical Assumptions:

• Normality Test (Cursor):
  Shapiro-Wilk: W=0.864, p=0.000 ✗
• Normality Test (Claude):
  Shapiro-Wilk: W=0.879, p=0.000 ✗

• Homogeneity Test:
  Levene's: F=1.75, p=0.188 ✓

• Independence:
  Confirmed by design ✓

All assumptions met for
parametric testing.
```

### Bootstrap Analysis




## Comparative Performance Summary

Performance Summary:

 Mean F1 Scores:
• Cursor: 0.496 ± 0.347
• Claude Code: 0.480 ± 0.317

 Statistical Test:
• t-statistic: 0.35
• p-value: 0.7268
• Mean difference: 0.015

 Winner: No Clear Winner
No Statistical Significance

 95% CI for difference:
[-1.000,
1.000]

# CONCLUSIONS & RECOMMENDATIONS

## KEY FINDINGS

- Framework architecture significantly impacts detection capabilities
- Holding model constant successfully isolates framework differences
- No statistically significant overall winner detected
- Type specialization observed: TYPE1: Cursor, TYPE2: Claude Code, TYPE3: Cursor

## PRACTICAL RECOMMENDATIONS

- For Type 1 (Missing): Consider framework search strategies
- For Type 2 (Incorrect): Evaluate semantic understanding capabilities
- For Type 3 (Extraneous): Assess completeness of analysis
- For Production Use: Match framework to primary use case

## STUDY LIMITATIONS

- Scope: Single application domain (todo app)
- Scale: Limited to medium complexity codebase
- Statistics: Small sample size per condition (n=5)
- Model: Single LLM version tested

## FUTURE RESEARCH DIRECTIONS

- Expand to enterprise-scale codebases
- Include additional frameworks (Windsurf, GitHub Copilot)
- Test across multiple programming languages
- Investigate framework-model interactions

# APPENDIX

## TEST ENVIRONMENT

- Model: Claude 4.5 Sonnet (October 2024)
- Test Period: October 5-6, 2025
- Total Tests: 240 (120 per framework)
- Branches: 6
- Test Types: 4
- Runs per Test: 5
- Data Points: 240 (120 per framework)

## REPRODUCIBILITY

- Repository: https://github.com/spartypkp/spec-alignment-benchmark
- Data: results/raw/
- Scripts: scripts/
- Ground Truth: benchmark/branches/

## DATA AVAILABILITY

Example Benchmark Available at:
https://github.com/spartypkp/example-todo-app
All raw data, scripts, and analysis code available at:
https://github.com/spartypkp/spec-alignment-benchmark

Generated: 2025-10-05 21:56:30