Benchmark Results: Interpretation

QCardEst/QCardCorr

December 21, 2025

1 Overview

This document provides a detailed interpretation of the JOB-light and STATS benchmark results comparing QCardEst (cardinality estimation) and QCardCorr (cardinality correction) approaches across different classical post-processing layers.

The results reveal a clear pattern: **correction-based approaches (QCardCorr) consistently outperform estimation-based approaches (QCardEst)** across most classical layers and both benchmarks. However, the best-performing layers differ between the two benchmarks, with Threshold excelling in JOB-light while Rational and RationalLog dominate in STATS. This suggests that layer effectiveness is context-dependent and influenced by the specific characteristics of the query workload and baseline estimator quality.

2 JOB-light Benchmark Results

2.1 Overview

For JOB-light, the **Threshold** layer achieves the best correction performance with a mean error difference of **0.39**, representing a remarkable improvement over both baseline estimators. This performance is **6.37 times better than the PostgreSQL baseline** (2.48) and **3.47 times better than the MSCN baseline** (1.35). The second-best performer is **PlaceValueNeg8** with an error of **0.42**, which is **5.91 times better than PostgreSQL** and **3.22 times better than MSCN**.

When examining all correction approaches, we observe that in **6 out of 8 cases where cardinality correction outperforms PostgreSQL, it also outperforms MSCN**. This pattern suggests that the quantum correction mechanisms are capable of systematically improving upon both classical estimators, with the improvement being more pronounced relative to PostgreSQL's estimates.

2.2 Why Threshold Correction Wins

The **threshold** layer achieves the best correction performance with a mean error difference of **0.39**, outperforming all other layers including the MSCN baseline (1.35) and PostgreSQL baseline (2.48).

2.2.1 Mechanism

The threshold layer (`SecondValueThreshold`) uses a gating mechanism based on ReLU activations with a threshold at 0.25:

```
posChange = 1 + ReLU(x[0] - 0.25) * x[1] * scalar^2
negChange = 1 + ReLU(x[2] - 0.25) * x[3] * scalar^2
result = posChange - negChange
```

2.2.2 Why It Works Well for Correction

1. **Selective Application**: The threshold mechanism only applies corrections when the quantum output exceeds 0.25, effectively filtering out noise and only making corrections when there is sufficient confidence in the quantum model's output.

2. **Bidirectional Corrections**: By combining positive and negative changes through subtraction, the threshold layer can handle both overestimations and underestimations from the baseline (PostgreSQL) estimator.

3. **Stability**: The threshold acts as a regularization mechanism, preventing the model from making unnecessary corrections when the baseline is already accurate. This is particularly important for correction tasks where many queries may already have reasonable estimates.

4. **Baseline Leverage**: Correction approaches benefit from leveraging existing knowledge (PostgreSQL's estimates) and only correcting when necessary. The threshold mechanism aligns perfectly with this philosophy by only activating corrections above a confidence threshold.

The threshold layer's success in correction (0.39 error) compared to estimation (5.19 error) demonstrates that **correction is fundamentally easier than full estimation** when a reasonable baseline exists.

2.3 Why PlaceValueNeg Variants Matter

The **PlaceValueNeg** variants show interesting and contrasting behaviors between estimation and correction:

- **PlaceValueNeg (estimation)**: 12.97 error - the worst performing layer for estimation

- **PlaceValueNeg (correction)**: 1.54 error - moderate performance for correction
- **PlaceValueNeg8 (estimation)**: 8.62 error - still poor for estimation
- **PlaceValueNeg8 (correction)**: 0.42 error - second best for correction after threshold

2.3.1 Mechanism

PlaceValueNeg uses a weighted sum where values are encoded as powers of (1 + scalar):

```
factors = [(1+scalar), (1+scalar)¹, ..., (1+scalar)^(n/2-1)]
if negativ:
    factors = [factors, -factors]  # Includes both positive and negative
result = sum(x * factors)
```

2.3.2 Why They Matter

1. **Representational Capacity**: The place-value encoding allows the model to represent corrections at different scales. The negative variant is crucial because it enables **bidirectional corrections** - the model can both increase and decrease the baseline estimate.
2. **Scale Sensitivity**: For estimation tasks, the place-value system struggles because it needs to represent absolute cardinalities spanning many orders of magnitude. However, for correction, the system only needs to represent **multiplicative factors** (typically close to 1.0), which is a much easier problem.
3. **Granularity Trade-off**: PlaceValueNeg8 (with 8 inputs) performs better than PlaceValueNeg (with 4 inputs) for correction (0.42 vs 1.54), showing that more granular encoding helps. However, this comes at a cost for estimation where PlaceValueNeg8 still performs poorly (8.62 vs 12.97).
4. **Correction Symmetry**: The negative factors in PlaceValueNeg variants allow symmetric corrections - the model can represent both “the baseline is 2× too high” and “the baseline is 2× too low” with similar representational complexity. This symmetry is less important for absolute estimation where values are always positive.

The dramatic improvement of PlaceValueNeg8 for correction (from 8.62 to 0.42) compared to estimation highlights that **the correction task is well-suited to multiplicative place-value representations**.

2.4 JOB-light Performance Summary

2.4.1 Correction (QCardCorr) Ranking:

1. **threshold**: 0.39 Best (6.37× better than PostgreSQL, 3.47× better than MSCN)
2. PlaceValueNeg8: 0.42 (5.91× better than PostgreSQL, 3.22× better than MSCN)
3. thresholdRatio: 0.49
4. linear: 0.45
5. rational: 0.99
6. PlaceValue: 0.61
7. rationalLog: 1.99
8. PlaceValueNeg: 1.54
9. PlaceValue8: 6.28 (outlier - see explanation below)

2.4.2 Estimation (QCardEst) Ranking:

1. **linear**: 1.43 Best (1.74× better than PostgreSQL)
2. PlaceValue8: 1.78 (1.40× better than PostgreSQL)
3. thresholdRatio: 3.15 (worse than PostgreSQL baseline)
4. rational: 3.44 (worse than PostgreSQL baseline)
5. threshold: 5.19 (worse than PostgreSQL baseline)
6. PlaceValue: 8.12 (worse than PostgreSQL baseline)
7. PlaceValueNeg8: 8.62 (worse than PostgreSQL baseline)
8. rationalLog: 4.38 (worse than PostgreSQL baseline)
9. PlaceValueNeg: 12.97 (worse than PostgreSQL baseline)

2.4.3 Key Observations for JOB-light

- **Correction dominance**: All correction approaches except PlaceValue8 outperform both PostgreSQL and MSCN baselines. This demonstrates the effectiveness of leveraging baseline estimates and applying selective corrections.

- **Estimation limitations**: For direct cardinality estimation, only **Linear** and **PlaceValue8** perform better than the PostgreSQL baseline. All other layers fail to improve upon PostgreSQL's estimates, highlighting the difficulty of absolute cardinality estimation compared to correction.
- **PlaceValue8 anomaly**: PlaceValue8 shows dramatically different behavior between correction (6.28 error, worst performer) and estimation (1.78 error, second-best). This stems from a fundamental limitation: PlaceValue8 can **only produce positive numbers**, meaning it can only correct cardinalities by **increasing them**. Since PostgreSQL estimates often need downward corrections, PlaceValue8 is unable to make the necessary adjustments, leading to poor correction performance. However, for estimation, this constraint is less problematic since cardinalities are inherently positive values.
- **PlaceValueNeg8 contrast**: The variant PlaceValueNeg8, which allows for negative numbers, performs excellently for correction (0.42, second-best) but poorly for estimation (8.62). This inverse relationship between PlaceValue8 and PlaceValueNeg8 highlights the importance of **bidirectional correction capability** for improving baseline estimates, while simpler positive-only representations work better for absolute value estimation.

3 STATS Benchmark Results

3.1 Overview

For STATS, the performance landscape differs significantly from JOB-light. The best correction performance is achieved by **RationalLog** with a mean error difference of **0.32**, which is **8.66 times better than the PostgreSQL baseline** (2.77). The second-best is **Rational** with an error of **1.04**, representing a **2.67 times improvement** over PostgreSQL. Notably, the Threshold layer, which dominated JOB-light, achieves 1.96 error (still better than PostgreSQL, but not the best).

This shift in optimal layer choice between benchmarks suggests that **the effectiveness of correction layers is sensitive to query workload characteristics**. Rational-based layers appear better suited to the STATS query patterns, possibly due to their ability to model multiplicative relationships more effectively in this context.

3.2 Key Findings

The STATS benchmark reveals several important patterns:

1. **Rational layers excel**: RationalLog and Rational achieve the best correction performance, with RationalLog showing an exceptional 8.66× im-

provement over PostgreSQL. This represents the largest improvement factor observed across both benchmarks.

2. **Linear leads estimation**: As in JOB-light, Linear maintains its position as the best estimation layer (1.65 vs 1.43 in JOB-light), and along with PlaceValue8 (2.97), are the only layers that outperform the PostgreSQL baseline for direct estimation.

3. **PlaceValue8 limitation confirmed**: PlaceValue8 again shows poor correction performance (13.65 error, worst among all correction approaches), confirming that its inability to produce negative corrections severely limits its effectiveness. This consistent pattern across both benchmarks validates the hypothesis that bidirectional correction capability is essential.

4. **Higher baseline error**: PostgreSQL baseline error is 2.77 for STATS vs 2.48 for JOB-light, indicating that STATS queries may be inherently more challenging for classical estimators, yet still allowing for significant quantum-based improvements.

5. **Correction vs Estimation gap**: The gap between best correction (0.32) and best estimation (1.65) in STATS is 1.33, which is larger than the gap in JOB-light (0.39 vs 1.43 = 1.04). This suggests correction has even greater advantage in STATS, despite the benchmark being more challenging overall.

3.3 STATS Performance Summary

3.3.1 Correction (QCardCorr) Ranking:

1. **rationalLog**: 0.32 Best (8.66× better than PostgreSQL)

2. **rational**: 1.04 (2.67× better than PostgreSQL)

3. threshold: 1.96

4. PlaceValueNeg8: 2.11

5. thresholdRatio: 2.53

6. PlaceValueNeg: 2.53

7. linear: 2.73

8. PlaceValue: 3.38

9. PlaceValue8: 13.65 (worst - see explanation below)

3.3.2 Estimation (QCardEst) Ranking:

1. **linear**: 1.65 Best (1.68× better than PostgreSQL)
2. PlaceValue8: 2.97 (slightly worse than PostgreSQL 2.77)
3. threshold: 4.54 (worse than PostgreSQL baseline)
4. thresholdRatio: 5.02 (worse than PostgreSQL baseline)
5. PlaceValue: 8.26 (worse than PostgreSQL baseline)
6. rational: 12.73 (worse than PostgreSQL baseline)
7. PlaceValueNeg8: 10.37 (worse than PostgreSQL baseline)
8. PlaceValueNeg: 13.23 (worse than PostgreSQL baseline)
9. rationalLog: 13.77 (worse than PostgreSQL baseline)

3.4 STATS-Specific Observations

1. **Rational layer dominance**: The exceptional performance of RationalLog (0.32) and Rational (1.04) for correction in STATS contrasts sharply with their moderate performance in JOB-light (1.99 and 0.99 respectively). This suggests that rational-based layers excel when dealing with certain types of query patterns or error distributions present in STATS.

2. **PlaceValue8 confirmed limitation**: PlaceValue8 again shows the worst correction performance (13.65), worse even than the PostgreSQL baseline (2.77). This confirms that **the inability to produce negative corrections is a fundamental limitation** for this layer. In contrast, PlaceValueNeg8, which can produce negative values, achieves 2.11 error, demonstrating that bidirectional correction is essential.

3. **Estimation remains challenging**: As in JOB-light, only Linear outperforms PostgreSQL for direct estimation, with PlaceValue8 performing slightly worse (2.97 vs 2.77). All other layers significantly underperform the baseline, reinforcing that absolute cardinality estimation is fundamentally more difficult than correction.

4. **Threshold versatility**: While Threshold is not the best in STATS, it still performs well (1.96, ranking 3rd) and outperforms the baseline. This demonstrates that Threshold is a robust, general-purpose choice, even if not always optimal.

Metric	JOB-light	STATS	Difference
Best Correction (threshold/rationalLog)	0.39	0.32	-0.07 (STATS better)
Best Estimation (linear)	1.43	1.65	+0.22 (STATS harder)
PostgreSQL Baseline	2.48	2.77	+0.29 (STATS harder)
Correction Advantage	1.04	1.33	+0.29 (larger gap)

4 Comparison: JOB-light vs STATS

4.1 Absolute Performance Differences

4.2 Key Differences

1. **Error Magnitude**: STATS shows consistently higher errors across all layers, suggesting the benchmark is inherently more challenging. This could be due to:
   - More complex query patterns
   - Different data distributions
   - Larger cardinality ranges
   - More join combinations
2. **Correction Advantage**: The advantage of correction over estimation is actually larger in STATS (1.33) compared to JOB-light (1.04). This suggests that:
   - Correction approaches can achieve even larger improvements in STATS (RationalLog: 8.66× better than PostgreSQL) compared to JOB-light (Threshold: 6.37× better)
   - The best correction in STATS (0.32) is actually better than the best correction in JOB-light (0.39), despite STATS being more challenging overall
   - Different layer architectures (RationalLog vs Threshold) are optimal for different workloads
3. **Layer Robustness**: The relative ranking of layers shows some variation between benchmarks:
   - **Correction**: Threshold excels in JOB-light while RationalLog dominates STATS, though PlaceValueNeg8 remains strong in both
   - **Estimation**: Linear and PlaceValue8 maintain their leading positions in both benchmarks
   - This variation suggests that optimal layer choice should consider workload characteristics

4. **Baseline Quality**: Both benchmarks have similar PostgreSQL baseline errors (2.48 vs 2.77), yet correction approaches achieve larger improvements in STATS (RationalLog: 8.66×) compared to JOB-light (Threshold: 6.37×), demonstrating that optimal layer choice can unlock significant correction potential.

4.3 Similarities

1. **Linear estimation dominance**: Linear layer consistently performs best for estimation in both benchmarks (1.43 JOB-light, 1.65 STATS), validating its simplicity and effectiveness.

2. **PlaceValueNeg8 reliability**: PlaceValueNeg8 consistently ranks high for correction (2nd in JOB-light, 4th in STATS), showing its reliability across different workloads.

3. **Estimation challenges**: Only Linear and PlaceValue8 consistently outperform PostgreSQL for estimation, while most other layers struggle with absolute value estimation in both benchmarks.

4. **PlaceValue8 constraint**: PlaceValue8 consistently fails at correction (6.28 JOB-light, 13.65 STATS) due to its inability to produce negative corrections, while performing well at estimation (1.78 JOB-light, 2.97 STATS).

5 When to Pick Which Layer: Rule of Thumb

Based on the combined results from both benchmarks, here is a practical guide for selecting classical layers. **Note that the optimal choice depends on the benchmark/workload characteristics**, with Threshold excelling in JOB-light but RationalLog dominating in STATS.

5.1 For Correction Tasks (QCardCorr - rowFactor)

For JOB-light workloads: Threshold

- Best performance (0.39, 6.37× better than PostgreSQL)
- Robust and interpretable selective correction mechanism
- Use when: Working with JOB-light-style queries and PostgreSQL baseline

For STATS workloads: RationalLog

- Best performance (0.32, 8.66× better than PostgreSQL)
- Exceptional improvement factor

- Use when: Working with STATS-style queries and need maximum correction accuracy

Alternative: Rational

- Good in STATS (1.04, 2.67× better than PostgreSQL)
- Moderate in JOB-light (0.99)
- Use when: You want a rational-based approach that works across benchmarks

Alternative: PlaceValueNeg8

- Consistently strong (0.42 JOB-light, 2.11 STATS)
- Second-best in JOB-light (5.91× better than PostgreSQL)
- Use when: You need fine-grained bidirectional correction control

Avoid for correction:

- **PlaceValue8**: Severely limited by inability to produce negative corrections (6.28 JOB-light, 13.65 STATS, worse than baseline in STATS). Can only increase cardinalities, making it unsuitable for correction tasks where downward adjustments are needed.
- **PlaceValue (non-negative variants)**: Moderate performance (0.61-3.38) but generally outperformed by layers with bidirectional capabilities.
- **Linear**: Moderate but not competitive for correction (0.45 JOB-light, 2.73 STATS), though it excels at estimation.

5.2 For Estimation Tasks (QCardEst - rows)

Primary Recommendation: linear

- Best performance for estimation in both benchmarks (1.43 JOB-light, 1.65 STATS)
- Simple and efficient
- Use when: You must do direct estimation without a baseline

Alternative: PlaceValue8

- Good performance (1.78 JOB-light, 2.97 STATS)
- More expressive than linear
- Use when: You need more representational capacity than linear provides

Avoid for estimation:

- PlaceValueNeg variants: Poor performance (8.62-13.23) - negative factors add complexity without benefit
- RationalLog: Poor performance (10.53 JOB-light, 12.72 STATS) - logarithmic ratio encoding doesn't work well for absolute values
- Threshold: Moderate (5.19 JOB-light, 4.54 STATS) but significantly worse than linear

5.3 General Principles

1. **Correction is easier than estimation**: If possible, use correction (QCardCorr) rather than estimation (QCardEst). Correction leverages baseline knowledge and only adjusts when needed. The advantage is significant: in JOB-light, best correction (0.39) is 3.7× better than best estimation (1.43); in STATS, best correction (0.32) is 5.2× better than best estimation (1.65).
2. **Workload matters**: Different layers excel for different query workloads. Threshold dominates JOB-light correction, while RationalLog excels in STATS. This suggests that layer selection should consider the specific characteristics of the target workload.
3. **Bidirectional correction is essential**: The PlaceValue8 vs PlaceValueNeg8 comparison demonstrates that the ability to produce negative corrections is crucial. PlaceValue8 (positive-only) fails at correction (6.28-13.65 error) but performs well at estimation (1.78-2.97), while PlaceValueNeg8 (bidirectional) excels at correction (0.42-2.11) but struggles with estimation (8.62-10.37).
4. **Only Linear and PlaceValue8 beat PostgreSQL for estimation**: For direct cardinality estimation, only these two layers consistently outperform the PostgreSQL baseline. All other layers fail to improve upon classical estimators for absolute value estimation, highlighting the difficulty of this task.
5. **PlaceValue8's fundamental limitation**: PlaceValue8 can only produce positive numbers, meaning it can only correct cardinalities by increasing them. Since PostgreSQL estimates often require downward corrections, PlaceValue8 is fundamentally unsuitable for correction tasks. However, this same constraint makes it effective for estimation where values are inherently positive.
6. **Baseline quality and correction advantage**: Correction approaches show dramatic improvements (6-8×) over PostgreSQL baselines, demonstrating that leveraging existing estimates and applying selective quantum corrections is highly effective. The improvement over MSCN (3-4×) is also substantial, showing that quantum corrections can improve even upon advanced learned estimators.

7. **Layer-task matching is critical**: The same layer can perform dramatically differently for correction vs estimation. Understanding the task requirements (absolute values vs. relative corrections) is essential for selecting the appropriate layer architecture.

6 Conclusions

The results demonstrate several key findings:

1. **Correction consistently outperforms estimation** across both benchmarks, with correction showing 3.7-5.2× improvement over the best estimation approaches. This demonstrates the clear advantage of leveraging baseline estimates.

2. **Optimal layer choice is workload-dependent**: Threshold excels in JOB-light (6.37× better than PostgreSQL), while RationalLog dominates STATS (8.66× better than PostgreSQL). This suggests that different query patterns benefit from different correction mechanisms.

3. **Bidirectional correction capability is essential**: The dramatic difference between PlaceValue8 (positive-only, fails at correction) and PlaceValueNeg8 (bidirectional, excels at correction) demonstrates that the ability to make both upward and downward corrections is fundamental to correction success.

4. **Only two layers beat PostgreSQL for estimation**: Linear and PlaceValue8 are the only layers that consistently outperform the PostgreSQL baseline for direct cardinality estimation, highlighting the difficulty of absolute value estimation compared to correction.

5. **PlaceValue8's constraint explains its behavior**: The observation that PlaceValue8 can only produce positive numbers directly explains why it fails at correction (where downward adjustments are needed) but succeeds at estimation (where values are inherently positive).

6. **Correction shows massive improvements**: The best correction approaches achieve 6-8× improvements over PostgreSQL and 3-4× improvements over MSCN, demonstrating the significant potential of quantum-enhanced correction mechanisms when combined with classical baselines.

7. **Layer-task matching matters**: The same layer architecture can show dramatically different performance for correction vs. estimation tasks, emphasizing that layer selection must consider the specific task requirements and constraints.