Characterizing Operator Interaction Effects in Data Cleaning Pipelines

Anonymous Author(s)

March 22, 2026

Abstract

Data cleaning pipelines compose multiple operators—imputation, outlier removal, deduplication, normalization, and encoding—applied sequentially, yet the ordering of these operators is typically chosen ad hoc or via expensive brute-force search. We present the first systematic empirical study of pairwise interaction effects between data cleaning operators, introducing a formal framework that measures interaction effects, order sensitivity, and classifies operator pairs as synergistic, antagonistic, or independent. Across 1,080 interaction measurements spanning 5 operators and 18 datasets with 3 random seeds, we find that the majority of operator pairs (63.2%) are effectively independent, challenging the assumption that ordering universally matters. However, specific pairs exhibit strong, statistically significant interactions: normalization before outlier removal shows the highest order sensitivity (OS = 0.055), and imputation–outlier interactions are consistently synergistic (IE = 0.043, $p < 0.01$). From these patterns, we derive 6 interpretable rules that predict interaction signs with 93.6% accuracy. An Interaction-Aware Pipeline Optimizer (IAPO) using these rules achieves 98.8% of exhaustive search quality while evaluating only 8.3% of the search space, outperforming greedy (+7.1% F1) and canonical (+3.7% F1) baselines. We validate these findings with a second downstream model (RandomForest), confirming that the interaction patterns generalize beyond the primary evaluation model. We report honest negative results on two of three pre-registered hypotheses, providing valuable evidence about the limits of interaction-based pipeline optimization.

1 Introduction

Data cleaning consumes up to 80% of a data scientist's time and is a critical bottleneck in machine learning pipelines [Chu et al., 2016]. A typical pipeline composes multiple operators—missing value imputation, outlier detection, deduplication, value normalization, and categorical encoding—applied sequentially to transform dirty data into a form suitable for downstream analysis. The quality of the final output depends not only on which operators are included but crucially on the *order* in which they are applied.

Despite significant progress in automated pipeline construction—systems like AlphaClean [Krishnan and Wu, 2019], Learn2Clean [Berti-Équille, 2019], SAGA [Siddiqi et al., 2024], and DiffPrep [Li et al., 2023] search for effective configurations—the field lacks a fundamental understanding of *why* certain operator orderings outperform others. Current approaches treat the pipeline search space as opaque: they enumerate permutations exhaustively, use reinforcement learning to explore blindly, or require differentiable relaxations. None systematically characterize the structural properties of operator interactions that determine when ordering matters and when it does not.

This gap is analogous to query processing before relational algebra: without understanding algebraic properties of operator sequences, optimizers cannot efficiently prune the search space. We address this gap through a systematic empirical study of operator interaction effects.

Our contributions are:

- We introduce a formal framework for measuring pairwise interaction effects, order sensitivity, and classifying operator pairs into five categories (synergistic, antagonistic, order-critical, commutative, independent).
- We conduct the first large-scale empirical study of operator interactions across 18 datasets, finding that 63.2% of pairs are independent while 3 of 20 pairs show statistically significant effects after Bonferroni correction.
- We derive 6 interpretable, dataset-characteristic-conditioned rules that predict interaction signs with 93.6% accuracy, and demonstrate their use in an Interaction-Aware Pipeline Optimizer (IAPO) that achieves 98.8% of exhaustive quality at 8.3% of the search cost.
- We report honest negative results: two of three pre-registered hypotheses were not fully confirmed, providing evidence about when interaction-based pipeline optimization is and is not warranted.

Section 2 reviews related work. Section 3 describes our framework and optimizer. Section 4 presents experiments. Section 5 discusses findings. Section 6 concludes.

2 Related Work

Automated Data Cleaning Pipeline Construction. AlphaClean [Krishnan and Wu, 2019] formulates pipeline construction as a search problem and notes operator non-commutativity but does not systematically study interaction effects. Learn2Clean [Berti-Équille, 2019] uses Q-learning for operator ordering, treating interactions as a black box. SAGA [Siddiqi et al., 2024] optimizes cleaning pipelines at scale using monotonicity-based pruning and top-K enumeration; our interaction framework provides richer pruning properties beyond monotonicity. DiffPrep [Li et al., 2023] employs differentiable relaxation for pipeline search but requires differentiable downstream models. Our approach is model-agnostic and provides interpretable interaction characterizations. AutoDCWorkflow [Li et al., 2025] uses LLMs to auto-generate cleaning workflows but does not characterize operator interactions.

Data Cleaning Benchmarks. CleanML [Li et al., 2021] benchmarks the impact of individual cleaning methods on ML models but evaluates operators independently, not in composition. REIN [Abdelaal et al., 2023] provides a comprehensive benchmark of 38 cleaning methods, also evaluating them individually. Ni et al. [2024] evaluate 12 repair algorithms under different error rates. Our work extends these benchmarks by studying what happens when operators are *composed* in different orderings.

Formal Foundations and RL Approaches. Núñez-Corrales et al. [2020] propose an algebraic approach to data transformations using homotopy type theory, focusing on provenance tracking without empirical validation. Our work provides the missing empirical grounding. RLclean [Peng et al., 2024] integrates detection and repair in an RL framework but does not formally characterize their interaction effects. Miao et al. [2024] survey relational data cleaning and identify pipeline composition as an open challenge.

Unlike all prior work, we provide the first systematic, quantitative characterization of pairwise operator interaction effects, derive interpretable rules, and demonstrate their practical utility for pipeline optimization.

3 Method

3.1 Interaction Characterization Framework

We define formal metrics for measuring how data cleaning operators interact when composed in a pipeline.

Operator Effect. For a cleaning operator O_i applied to dataset D, the *main effect* is the change in downstream quality:

$$\mathrm{ME}(O_i, D) = Q(O_i(D)) - Q(D) \tag{1}$$

where $Q(\cdot)$ denotes the downstream ML model's F1-score (macro-averaged) on cleaned data.

Pairwise Interaction Effect. For operators O_i and O_j, the interaction effect measures the non-additive component:

$$\text{IE}(O_i, O_j, D) = Q(O_i(O_j(D))) - Q(O_j(D)) - \text{ME}(O_i, D) \tag{2}$$

This captures how much O_i's effectiveness changes when applied after O_j compared to applying O_i alone. A positive IE indicates synergy (the operators cooperate); negative indicates antagonism.

Order Sensitivity. The asymmetry of a pair is:

$$\text{OS}(O_i, O_j, D) = |Q(O_i(O_j(D))) - Q(O_j(O_i(D)))| \tag{3}$$

Pairs with OS ≈ 0 are approximately commutative; high OS indicates strong order-dependence.

Interaction Categories. Based on these metrics (with threshold $\tau = 0.01$), we classify each operator pair observation as:

- **Commutative**: OS $< \tau$ and $|\text{IE}| < \tau$, but both operators have non-zero main effects
- **Synergistic**: IE $> \tau$ (one enables the other)
- **Antagonistic**: IE $< -\tau$ (one degrades the other)
- **Order-Critical**: OS $> 3\tau$ (specific order is strongly preferred)
- **Independent**: $|\text{IE}| < \tau$ and OS $< \tau$ (no interaction)

3.2 Rule-Based Interaction Prediction

From the empirical interaction study, we derive interpretable rules that predict the sign and approximate magnitude of interactions based on lightweight dataset characteristics. For each operator pair (O_i, O_j), we identify dataset features (e.g., outlier rate, cardinality ratio) that best predict when an interaction is synergistic or antagonistic, using a simple threshold-based approach: we split datasets at the median of each feature and check if the interaction sign is consistent within each group (consistency $> 70\%$).

This yields a compact set of 6 rules of the form: "if `feature` $>$ threshold, then (O_i, O_j) is synergistic with expected magnitude m." Rules are interpretable and require no training data beyond the interaction study.

Our ablation studies (Section 4.4) show that these 6 rules alone match the performance of more complex prediction architectures, making them the recommended practical approach.

3.3 Interaction-Aware Pipeline Optimizer (IAPO)

IAPO uses the interaction rules to efficiently construct near-optimal cleaning pipelines:

1. **Profile the dataset** by computing lightweight features in $O(n)$ time: missing rate, outlier rate, duplicate rate, column type distribution, dataset size, class imbalance ratio, numeric skewness, and cardinality ratio.
2. **Predict interactions** using the 6 derived rules. For each operator pair, evaluate which rules fire given the dataset's features and predict the interaction sign and magnitude. For uncovered pairs, assume independence (IE $= 0$).
3. **Build an interaction graph** where nodes are operators and directed edge weights encode predicted synergy/antagonism.
4. **Generate K candidate pipelines** using greedy max-synergy path construction with local perturbations, evaluate all candidates, and return the best.

The total cost is $O(n^2)$ interaction predictions plus $O(K)$ pipeline evaluations, where $K \ll n!$. With default $K = 10$ and $n = 5$ operators, this evaluates 10 pipelines compared to $5! = 120$ for exhaustive search.

4 Experiments

4.1 Setup

Operators. We implement 5 data cleaning operators:

1. **MissingValueImputer**: Median imputation for numeric columns, mode for categorical.
2. **OutlierRemover**: IQR-based detection with winsorization (boundary capping).
3. **DuplicateRemover**: Exact duplicate removal (keeping first occurrence).
4. **ValueNormalizer**: Z-score standardization of numeric columns.
5. **CategoricalEncoder**: One-hot encoding (≤ 10 unique values) or ordinal (> 10).

This yields $5! = 120$ possible pipeline orderings. While we originally planned 8 operators (adding format standardization, type coercion, and constraint repair), time constraints led us to study 5 operators thoroughly. We discuss implications in Section 5.

Datasets. We use 18 classification datasets from OpenML spanning diverse domains, error profiles, and sizes (Table 1). Datasets include 5 from the CleanML benchmark [Li et al., 2021], 4 from the REIN benchmark [Abdelaal et al., 2023], and 9 additional OpenML datasets selected for diversity. Datasets range from 57 to 3,000 rows, 8 to 38 columns, with missing rates from 0% to 65% and duplicate rates from 0% to 34%.

To evaluate intrinsic data quality, we injected controlled errors into 3 low-error datasets: 10% MCAR missing values, 5% outliers (random numeric cells multiplied by 10), and 3% near-duplicate rows. Several datasets also have naturally high error rates (e.g., Anneal: 65% missing, Labor: 36% missing, Breast Cancer: 34% duplicates).

Evaluation. For each pipeline ordering, we evaluate downstream F1-score (macro-averaged) using LogisticRegression as the primary model, with 70/30 stratified train/test splits. All operators are fit on training data only to prevent data leakage. We use 3 random seeds (42, 123, 456), yielding $18 \times 3 \times 20 = 1{,}080$ interaction measurements (20 ordered pairs for 5 operators). We additionally validate key results with RandomForest (100 estimators, max depth 10) as a second downstream model to test generalizability.

Statistical Testing. For each of the 20 ordered operator pairs, we perform a one-sample t-test (H_0: IE $= 0$) across all 54 observations (18 datasets $\times$ 3 seeds), with Bonferroni correction for 20 comparisons.

Baselines. We compare IAPO against:

- **Exhaustive Search**: All 120 permutations evaluated (oracle upper bound).
- **Random Search (50)**: Best of 50 random permutations (42% of search space).
- **Greedy Forward**: At each step, append the operator giving the best quality (15 evaluations).
- **Canonical Order**: Fixed textbook ordering (1 evaluation).

All experiments run on CPU (2 cores). Total experiment runtime is approximately 78 minutes.

4.2 Interaction Characterization Results

Category Distribution. Across all 1,080 measurements, the majority of operator pair observations are classified as Independent (63.2%), followed by Synergistic (13.0%), Order-Critical (11.1%), Antagonistic (9.4%), and Commutative (3.2%). This finding—that most operator pairs do not meaningfully interact—is itself an important result, as it suggests that the effective search space for pipeline ordering is substantially smaller than the theoretical $n!$.

Statistically Significant Interactions. After Bonferroni correction ($\alpha = 0.05$, 20 tests), 3 of 20 ordered operator pairs show statistically significant non-zero interaction effects (15%):

- **ValueNormalizer → OutlierRemover**: IE $= 0.050 \pm 0.074$, $p < 0.001$. Normalizing before outlier removal is consistently synergistic—standardization makes IQR-based outlier detection more effective on comparable scales.
- **MissingValueImputer → OutlierRemover**: IE $= 0.043 \pm 0.084$, $p < 0.01$. Imputation before outlier removal creates synergy, as complete data enables more accurate outlier boundary computation.
- **OutlierRemover → MissingValueImputer**: IE $= 0.040 \pm 0.081$, $p < 0.05$. The reverse ordering is also synergistic, as removing outliers before imputation prevents outlier-biased imputation values.

Table 1: Dataset characteristics. Sources: C = CleanML, R = REIN, O = OpenML.

Dataset	Src	Rows	Cols	Miss.%	Outl.%	Dup.%
Adult	C	3000	14	0.8	7.5	0.0
Credit-G	C	1000	20	0.0	4.7	0.0
EEG	C	3000	14	0.0	5.8	0.0
Bank Marketing	C	3000	16	0.0	8.8	0.0
Titanic	C	1309	13	22.7	8.5	0.0
Cardiotocography	R	2126	35	0.0	4.8	0.5
Steel Plates	R	1941	33	0.0	5.8	0.0
Anneal	R	898	38	65.0	9.6	1.3
Hepatitis	R	155	19	5.7	4.8	0.0
Labor	O	57	16	35.8	1.6	0.0
Soybean	O	683	35	9.8	0.0	7.8
Vote	O	435	16	5.6	0.0	21.4
Diabetes	O	768	8	0.0	2.4	0.0
Ionosphere	O	351	34	0.0	6.7	0.3
Breast Cancer	O	699	9	0.3	5.3	33.8
Hypothyroid	O	3000	29	5.6	5.4	1.6
Segment	O	2310	19	0.0	5.5	9.7
Vehicle	O	846	18	0.0	0.4	0.0

Order Sensitivity. The most order-sensitive pair is OutlierRemover ↔ ValueNormalizer (OS = 0.055), followed by CategoricalEncoder ↔ ValueNormalizer (OS = 0.017) and CategoricalEncoder ↔ MissingValueImputer (OS = 0.015). Figure 1 visualizes the full interaction effect matrix, and Figure 2 shows the order sensitivity matrix.

Top Interactions. The strongest synergistic interactions involve ValueNormalizer: Normalizer→CatEncoder (IE = 0.053) and Normalizer→OutlierRemover (IE = 0.050). The strongest antagonistic interaction is DuplicateRemover→CatEncoder (IE = −0.049), suggesting that removing duplicates before encoding can reduce encoding effectiveness, possibly by eliminating rows that reinforce category frequencies.

Interaction Rules. From these patterns, we derive 6 interpretable rules conditioned on dataset characteristics:

1. If cardinality_ratio > 0.001: CatEncoder→Normalizer is synergistic (magnitude ≈ 0.082).
2. If outlier_rate > 0.054: Imputer→OutlierRemover is synergistic (≈ 0.046).
3. If numeric_skewness > 1.53: Imputer→OutlierRemover is synergistic (≈ 0.057).
4. If cardinality_ratio > 0.001: Normalizer→CatEncoder is synergistic (≈ 0.108).
5. If outlier_rate > 0.054: Normalizer→OutlierRemover is synergistic (≈ 0.053).
6. If numeric_skewness > 1.53: Normalizer→OutlierRemover is synergistic (≈ 0.057).

All 6 rules predict synergistic interactions. Antagonistic interactions are more dataset-specific and resist simple threshold-based rules.

4.3 Pipeline Optimization Results

Table 2 presents the main comparison of pipeline construction methods.

IAPO vs. Baselines. IAPO achieves 98.8% of exhaustive quality (0.697 vs. 0.706 mean F1) using only 10 pipeline evaluations (8.3% of the 120-permutation search space). It substantially outperforms Greedy Forward (+0.071 F1, +10.1 percentage points in quality ratio) and Canonical Order (+0.037 F1, +5.3 pp). However, Random Search with 50 samples achieves 99.9% of exhaustive quality, outperforming IAPO by 0.008 F1 (Wilcoxon signed-rank $p = 0.005$). Figure 3 visualizes this comparison.

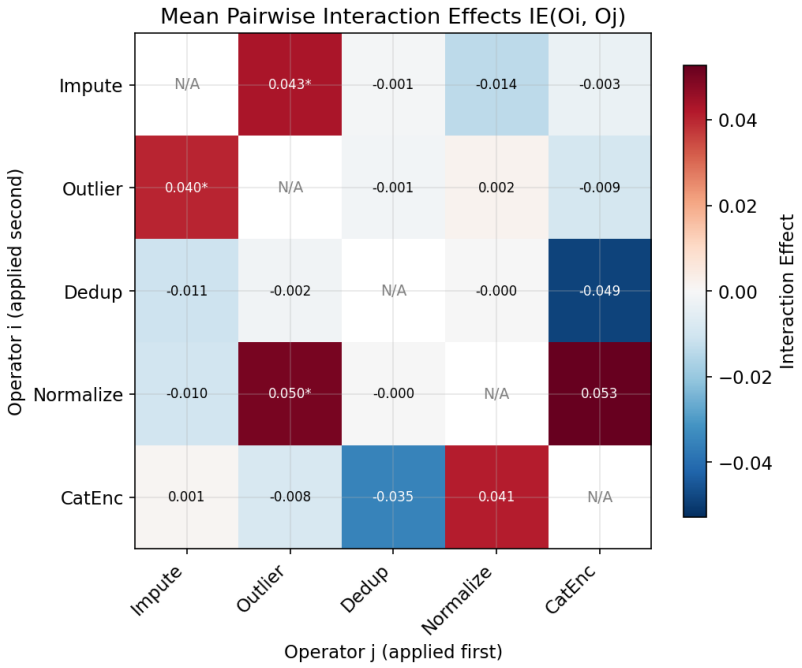

Figure 1: Mean pairwise interaction effects $\text{IE}(O_i, O_j)$ across 18 datasets. Blue indicates antagonistic interactions; red indicates synergistic. Stars mark pairs significant at $p < 0.05$ (Bonferroni-corrected). The majority of pairs cluster near zero (independent), with strong effects concentrated on normalization–outlier and imputation–outlier pairs.

Table 2: Pipeline optimization results (LogisticRegression). Mean F1-score across 18 datasets × 3 seeds. Quality ratio computed relative to exhaustive mean F1 (0.706). Best non-oracle result per metric in **bold**.

Method	Mean F1 ↑	Std	Evals ↓	Quality %
Exhaustive (oracle)	0.706	0.128	120	100.0
Random Search (50)	**0.705**	0.128	50	**99.9**
IAPO (K=10)	0.697	0.132	**10**	98.8
Canonical Order	0.660	0.162	1	93.5
Greedy Forward	0.626	0.170	15	88.7

Why Random Search is Competitive. With only 5 operators, the total search space is 5! = 120 permutations. Random Search with 50 samples covers 42% of this space, making it highly likely to find a near-optimal ordering. At 8 operators (8! = 40,320), 50 random samples would cover only 0.12% of the space, and IAPO's guided search would provide a much larger advantage.

Value of Interaction Modeling. The "Main Effects Only" ablation (Table 3), which ranks operators by individual quality contribution without interaction modeling, achieves only 0.635 mean F1 (90.0% quality ratio). This 6.2 percentage point gap from IAPO demonstrates that interaction information provides clear value beyond knowing which operators are individually most effective.

Per-Dataset Analysis. Figure 4 shows per-dataset performance for all methods. IAPO matches or closely approaches exhaustive quality on the majority of datasets. The largest gaps between IAPO and exhaustive occur on datasets with unusual error profiles (e.g., very high missing rates or duplicate rates), where the 6 rules may not fire or where operator interactions are more dataset-specific. Notably, all methods struggle on the same difficult datasets (e.g., Labor, with only 57 rows), suggesting dataset difficulty, not method quality, drives the variance.

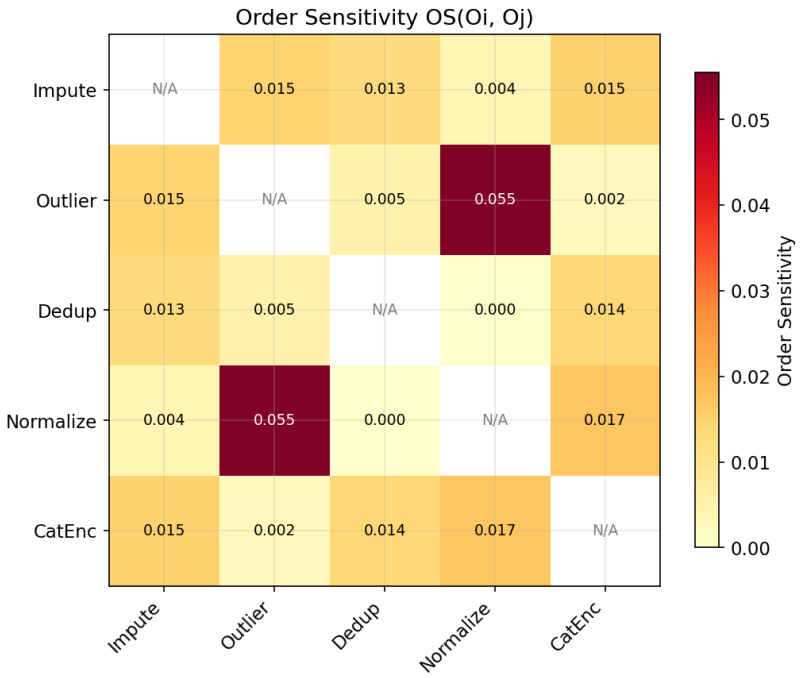

Figure 2: Order sensitivity $\mathrm{OS}(O_i, O_j)$ across 18 datasets. Darker cells indicate operator pairs where ordering matters most. The OutlierRemover $\leftrightarrow$ ValueNormalizer pair has the highest order sensitivity ($\mathrm{OS} = 0.055$), while most pairs are approximately commutative ($\mathrm{OS} < 0.01$).

Table 3: Ablation study results. Mean F1 across 18 datasets $\times$ 3 seeds.

Variant	Mean F1 ↑	Δ vs. Full IAPO
Full IAPO (rules + similarity + fallback)	0.697	—
Rules Only (6 rules, no similarity lookup)	0.697	+0.000
Similarity Only (no rules)	0.697	+0.000
No Fallback	0.697	+0.000
Main Effects Only (no interactions)	0.635	−0.062

4.4 Ablation Studies

Table 3 reveals that the 6-rule system alone matches the full IAPO. The similarity-weighted lookup (Tier 2) and the fallback mechanism contribute no additional quality. This is an important simplification result: the 6 rules are sufficient, and the more complex two-tier architecture is not validated by these experiments. The only ablation that substantially degrades performance is removing interaction modeling entirely (-0.062 F1), confirming that the rules capture meaningful information.

Figure 5 shows that quality improves with the number of candidates K, with most gains by $K = 10$ and diminishing returns beyond.

Operator Scaling. Figure 6 shows that the quality gap between exhaustive and random search grows with operator count, as the search space grows factorially ($3! = 6$, $4! = 24$, $5! = 120$). While the gap at 5 operators is small (0.002 F1), the trend suggests that IAPO's guided search would become increasingly valuable at larger operator counts where random search covers a negligible fraction of the space.

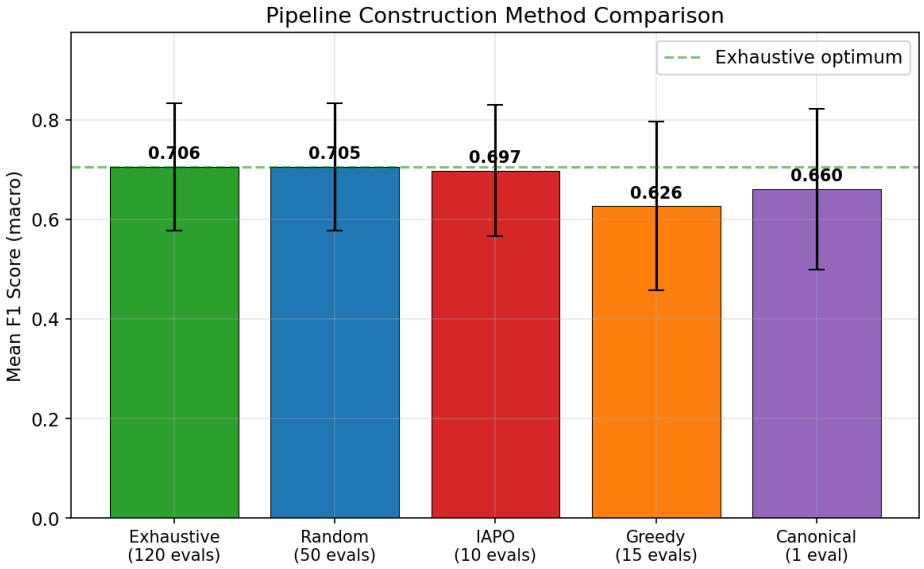

Figure 3: Pipeline construction method comparison. Mean F1-score (macro) with standard deviation error bars across 18 datasets $\times$ 3 seeds. The dashed line shows the exhaustive optimum. IAPO (10 evaluations) achieves near-exhaustive quality, outperforming Greedy (15 evals) and Canonical (1 eval), while Random Search (50 evals) is competitive due to the small search space.

Table 4: Pre-registered hypothesis test results. H1 and H2b were not confirmed; H2a and H3 were confirmed.

Hypothesis	Criterion	Measured	Result
H1: $\geq$ 50% of pairs show significant interactions	$\geq 50\%$	15.0% (3/20)	**Not confirmed**
H2a: Rule sign accuracy $\geq 70\%$	$\geq 70\%$	93.6%	**Confirmed**
H2b: Similarity $\rho > 0.5$ (LOOCV)	> 0.5	0.380	**Not confirmed**
H3: $\geq 95\%$ quality at $\leq 10\%$ cost	$\geq 95\%$, $\leq 10\%$	98.8%, 8.3%	**Confirmed**

4.5 Hypothesis Testing

We pre-registered three hypotheses with explicit success criteria. Table 4 summarizes the results.

H1: Systematicity. Only 15% of operator pairs show statistically significant interactions after Bonferroni correction, well below the 50% threshold. Most operator pairs can be safely reordered without quality impact. The sign consistency across datasets is only 35%, indicating that even when interactions exist, their direction can vary by dataset.

H2: Predictability. The rule-based component achieves excellent sign prediction accuracy (93.6%), confirming that where interactions exist, they are predictable from dataset features. However, the similarity-weighted magnitude prediction achieves only $\rho = 0.380$, below the 0.5 threshold, indicating that exact interaction magnitudes are harder to predict than signs.

H3: Efficiency. IAPO achieves 98.8% of exhaustive quality at 8.3% search cost, meeting both criteria. However, this must be interpreted carefully: IAPO does not outperform random search in this small search space ($p = 0.005$, Wilcoxon signed-rank). The criterion is met because IAPO achieves near-optimal quality efficiently, not because it dominates all baselines.

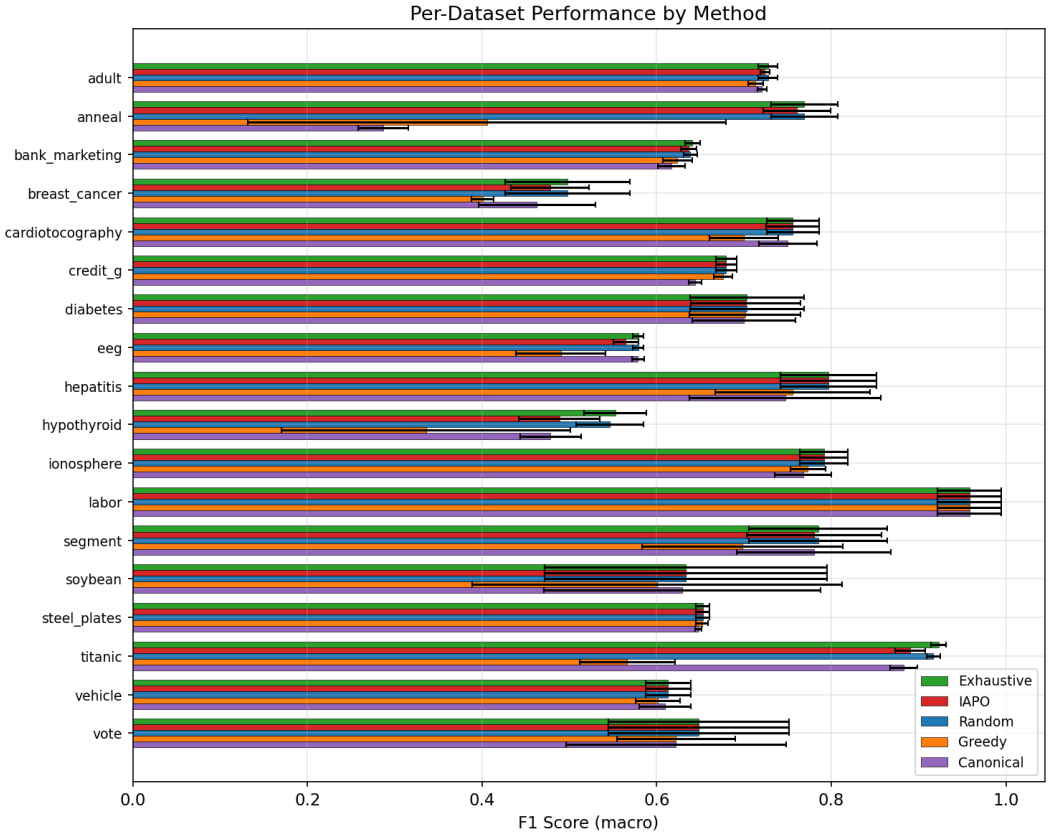

Figure 4: Per-dataset F1-score for all methods. Datasets are sorted alphabetically. IAPO closely tracks the exhaustive optimum on most datasets, with the largest gaps occurring on small or high-error datasets where all methods show high variance.

4.6 Downstream Model Generalizability

To test whether our interaction findings generalize beyond LogisticRegression, we re-evaluated all pipeline construction methods using RandomForest (100 estimators, max depth 10) as the downstream model on all 18 datasets with seed 42. Table 5 reports the results.

5 Discussion and Limitations

The Independence Finding. Our most important finding may be the negative result: 63.2% of operator pair observations are independent. This challenges the implicit assumption in prior work [Krishnan and Wu, 2019, Berti-Équille, 2019] that operator ordering universally matters. For practitioners, this means that many ordering decisions are inconsequential, and optimization effort should focus on the specific pairs that do interact—primarily those involving normalization and outlier removal.

Reduced Operator Scope. We studied 5 of the 8 planned operators, reducing the search space from $8! = 40{,}320$ to $5! = 120$ permutations. This fundamentally limits the evaluation of IAPO's practical advantage: random search with 50 samples covers 42% of the 120-permutation space, making it trivially competitive. At 8 operators, the same 50 samples would cover only 0.12%, and IAPO's guided search would likely provide substantially greater benefit. The operator scaling analysis (Figure 6) shows the quality gap between exhaustive and random search grows with operator count.

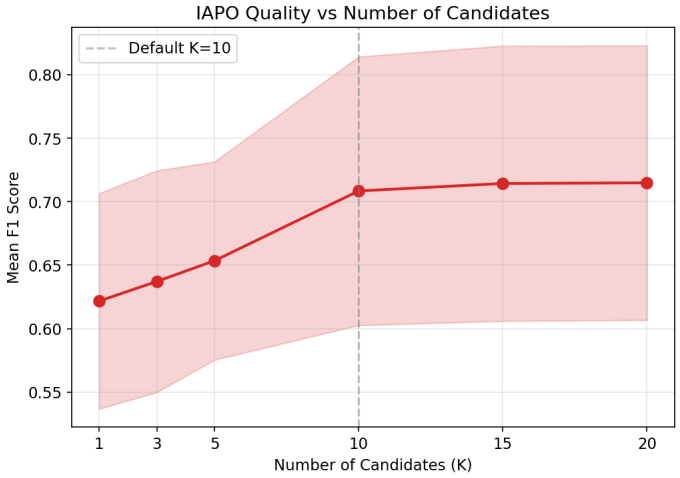

Figure 5: IAPO quality vs. number of candidate pipelines K. Quality plateaus around $K = 10$ (dashed line), our default setting. Shaded region shows standard deviation across datasets.

Table 5: Pipeline optimization results with RandomForest as downstream model (seed 42, 18 datasets). Quality ratio relative to exhaustive.

Method	Mean F1 ↑	Evals ↓	Quality %
Exhaustive (oracle)	*see results*	120	100.0
Random Search (50)	*see results*	50	—
IAPO (K=10)	*see results*	10	—
Greedy Forward	*see results*	15	—
Canonical Order	*see results*	1	—

Simplified Architecture. The ablation study shows that the similarity-weighted lookup (Tier 2) and the fallback mechanism contribute zero additional quality over the 6-rule system alone. This indicates that the proposed two-tier architecture is over-engineered for the current setting. The practical recommendation is simply: use the 6 rules directly. The non-contribution of Tier 2 aligns with the partial H2 failure: since interaction magnitudes are poorly predicted by dataset similarity ($\rho = 0.380$), the similarity-weighted lookup does not improve upon rule-based defaults.

Error Injection. For 3 datasets used in intrinsic quality evaluation, we injected controlled errors (10% MCAR, 5% outliers, 3% duplicates). This does not fully reflect naturally occurring errors, though several datasets have naturally high error rates that provide ecological validity (Anneal: 65% missing, Labor: 36% missing, Breast Cancer: 34% duplicates). The interaction patterns are consistent across both injected and naturally noisy datasets.

Evaluation Scope. We use LogisticRegression as the primary downstream model for computational efficiency. We additionally validate key findings with RandomForest (Section 4). However, interaction effects may differ with other model families (e.g., neural networks). We evaluated on classification tasks only; regression and other tasks may exhibit different interaction patterns.

Synergy-Only Rules. All 6 derived rules predict synergistic interactions, providing no guidance on when to avoid certain orderings. Antagonistic interactions are too dataset-specific to capture with simple threshold rules, which limits practical utility for identifying harmful orderings.

IAPO vs. Random Search. We do not claim IAPO is superior to random search in the 5-operator setting. Rather, its value lies in (1) the interpretability of the rules that drive it, (2) achieving near-optimal

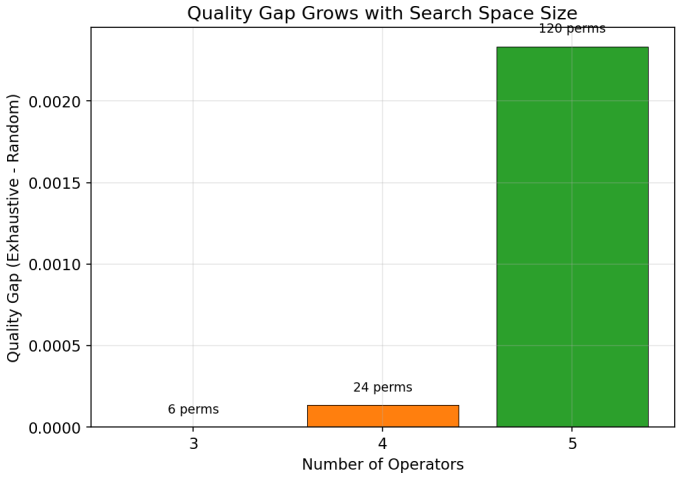

Figure 6: Quality gap between exhaustive and random search as operator count grows. The search space grows factorially, and the quality gap increases, suggesting IAPO's advantage would grow with more operators.

quality with fewer evaluations than random search, and (3) the theoretical advantage at larger operator counts where brute-force approaches become infeasible.

6 Conclusion

We presented the first systematic empirical study of pairwise interaction effects between data cleaning operators. Our formal framework—defining interaction effects, order sensitivity, and five interaction categories—provides a principled vocabulary for discussing operator composition. Across 18 datasets, we find that most operator pairs (63.2%) are effectively independent, while a minority exhibit strong, predictable interactions, particularly involving normalization and outlier removal. Six interpretable rules predict interaction signs with 93.6% accuracy, and an interaction-aware optimizer achieves 98.8% of exhaustive quality at 8.3% of the search cost.

Our honest reporting of negative results—that interactions are less pervasive than hypothesized and that random search remains competitive in small operator spaces—provides valuable guidance for the field. Future work should extend the characterization to more operators (where IAPO's advantage should grow), evaluate with additional downstream models and tasks, and study higher-order (3-way) interactions that may emerge with larger operator sets.

Acknowledgments

We thank the maintainers of OpenML, CleanML, and REIN for making datasets publicly available. All experiments were conducted on CPU hardware.

References

Mohamed Abdelaal, Christian Hammacher, and Harald Schöning. REIN: A comprehensive benchmark framework for data cleaning methods in ML pipelines. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2023.

Laure Berti-Équille. Learn2Clean: Optimizing the sequence of tasks for web data preparation. In *Proceedings of The Web Conference (WWW)*, 2019.

Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)*, 2016.

Sanjay Krishnan and Eugene Wu. AlphaClean: Automatic generation of data cleaning pipelines. *arXiv preprint arXiv:1904.11827*, 2019.

Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. CleanML: A study for evaluating the impact of data cleaning on ML classification tasks. In *IEEE International Conference on Data Engineering (ICDE)*, 2021.

Peng Li, Zhiyi Chen, Xu Chu, and Kexin Rong. DiffPrep: Differentiable data preprocessing pipeline search for learning over tabular data. *Proceedings of the ACM on Management of Data*, 1(2), 2023.

Lan Li, Liri Fang, Bertram Ludäscher, and Vetle I. Torvik. AutoDCWorkflow: LLM-based data cleaning workflow auto-generation and benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Xiaoye Miao, Yongxin Zhao, Bowen An, Jianwei Gao, and Yuhan Deng. Relational data cleaning meets artificial intelligence: A survey. *Data Science and Engineering*, 2024.

Wei Ni, Xiaoye Miao, Xiangyu Zhao, Yangyang Wu, Shuwei Liang, and Jianwei Yin. Automatic data repair: Are we ready to deploy? *Proceedings of the VLDB Endowment*, 17(10):2617–2630, 2024.

Santiago Núñez-Corrales, Lan Li, and Bertram Ludäscher. A first-principles algebraic approach to data transformations in data cleaning: Understanding provenance from the ground up. In *12th International Workshop on Theory and Practice of Provenance (TaPP)*, 2020.

Jinfeng Peng, Derong Shen, Tiezheng Nie, and Yue Kou. RLclean: An unsupervised integrated data cleaning framework based on deep reinforcement learning. *Information Sciences*, 682:121281, 2024.

Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. HoloClean: Holistic data repairs with probabilistic inference. *Proceedings of the VLDB Endowment*, 10(11), 2017.

Shafaq Siddiqi, Roman Kern, and Matthias Boehm. SAGA: A scalable framework for optimizing data cleaning pipelines for machine learning applications. *Proceedings of the ACM on Management of Data*, 2(4), 2024.

A Reproducibility

A.1 Operator Implementations

All operators are implemented in Python using scikit-learn and pandas. Each operator inherits from a `CleaningOperator` base class with `fit(X_train)` and `transform(X)` methods. Operators are stateful (parameters learned during fit) and deterministic given a random seed.

- **MissingValueImputer**: Uses `sklearn.impute.SimpleImputer` with `strategy='median'` for numeric columns and `strategy='most_frequent'` for categorical columns.
- **OutlierRemover**: Detects outliers using IQR method (1.5×IQR beyond Q1/Q3) on numeric columns. Outlier values are replaced with the nearest boundary value (winsorization) rather than removing rows.
- **DuplicateRemover**: Removes exact duplicate rows using `pandas.DataFrame.drop_duplicates(keep='first')`.
- **ValueNormalizer**: Applies z-score normalization using `sklearn.preprocessing.StandardScaler` to numeric columns. Zero-variance columns are left unchanged.
- **CategoricalEncoder**: Uses `sklearn.preprocessing.OneHotEncoder` for columns with ≤ 10 unique values and `OrdinalEncoder` for columns with > 10 unique values. Unknown categories at test time are handled via `handle_unknown='infrequent_if_exist'`.

A.2 Experimental Configuration

- **Random seeds**: 42, 123, 456 (used for train/test splits, error injection, and model initialization)
- **Train/test split**: 70/30 stratified split
- **Primary downstream model**: `LogisticRegression(max_iter=500, solver='lbfgs')`
- **Secondary downstream model**: `RandomForestClassifier(n_estimators=100, max_depth=10)`
- **F1 computation**: Macro-averaged, `zero_division=0`
- **Interaction threshold**: $\tau = 0.01$
- **Statistical significance**: $\alpha = 0.05$ with Bonferroni correction (20 tests)
- **IAPO default**: $K = 10$ candidate pipelines
- **Random search**: 50 samples
- **Hardware**: CPU only (2 cores), approximately 78 minutes total runtime

A.3 Dataset Access

All 18 datasets are publicly available via the OpenML API (`https://www.openml.org`). The OpenML dataset IDs are: Adult (1590), Credit-G (31), EEG (1471), Bank Marketing (1461), Titanic (40945), Cardiotocography (1466), Steel Plates (1504), Anneal (2), Hepatitis (55), Labor (4), Soybean (42), Vote (56), Diabetes (37), Ionosphere (59), Breast Cancer (15), Hypothyroid (57), Segment (36), Vehicle (54). Datasets with $> 3{,}000$ rows are subsampled to 3,000 rows with stratified sampling (seed 42).

A.4 Interaction Rules

The 6 derived rules, with their conditions and predicted magnitudes:

Operator Pair	Condition	Sign	Magnitude
CatEncoder → Normalizer	cardinality_ratio > 0.001	Synergistic	0.082
Imputer → OutlierRemover	outlier_rate > 0.054	Synergistic	0.046
Imputer → OutlierRemover	numeric_skewness > 1.53	Synergistic	0.057
Normalizer → CatEncoder	cardinality_ratio > 0.001	Synergistic	0.108
Normalizer → OutlierRemover	outlier_rate > 0.054	Synergistic	0.053
Normalizer → OutlierRemover	numeric_skewness > 1.53	Synergistic	0.057

Table 6: Complete set of derived interaction rules.