Appendix

Proof of Theorem 1 . Inspired by [1], we prove the NP-completeness by reducing the minimum vertex cover problem to the OSR problem.

Given a removal set I_N, whether $L(I\backslash I_N) > \beta$ can be verified in $O(mn^2)$. The time for verifying the non-conflict condition is $O(mn^2)$, and the time for checking minimality is $O(c^2)$. Therefore, whether a removal set I_N is a solution of the decision version of Problem 1 can be verified in polynomial time.

Consider a graph $G = (E, V)$, where $V = \{v_1, v_2, \ldots, v_{n_V}\}$ represents the set of vertices and $E = \{e_1, e_2, \ldots, e_{n_E}\}$ denotes the set of edges. This graph corresponds to a relation instance I with $R = \{E_1, E_2, \ldots, E_{n_E}, V, T, D\}$. Each edge $e_i = (v_1^i, v_2^i)$ gives rise to two tuples, t_1^i and t_2^i. For these tuples, we have $t_1^i[E_i] = t_2^i[E_i] = e$, $t_1^i[V] = g(v_1^i)$, and $t_2^i[V] = g(v_2^i)$. Additionally, $t_1^i[T] = u_1^i$ and $t_2^i[T] = u_2^i$. The remaining attribute values for t_1^i and t_2^i are set to 0. The function $g(\cdot)$ maps vertices $v \in V$ to positive numbers, such that $g(v_1^i) = id(v_1^i)b$ and $g(v_2^i) = id(v_2^i)b$. $id(v)$ are the index of $v \in V$. The inverse mapping $g^{-1}(\cdot)$ is defined as $g^{-1}(t_j^i[V]) = v_j^i$. The values u_j^i are computed as $u_1^i = (2i+1)^2 B$ and $u_2^i = (2i+2)^2 B$. Here, e, b, and B are positive numbers. The tuples induced by all $e_i \in E$ collectively form the set S_1.

A vertex $v_l \in V$ gives rise to a tuple t_l. For this tuple, we have $t_l[V] = t_l[T] = g(v_l)$, $t_l[D] = d$, and all other attribute values are set to 0. Here, $d > 0$ denotes a positive number. The collection of tuples derived from all $v_l \in V$ constitute the set S_2. Note that each vertex of v_l in this set is equal to some v_j^i vertices in S_1. These are the same vertices, represented differently. Furthermore, a tuple $t_j^i \in S_1$ corresponds to four additional tuples $t_{j1}^i, t_{j2}^i, t_{j3}^i, t_{j4}^i$, where $t_{jr}^i[V] = g(v_j^i) + r\triangle$ for $r = 1, 2, 3, 4$, and all other attributes are 0. The positive number $\triangle$ is a constant. The set of all such t_{jr}^i tuples is denoted as S_3. Each tuple $t_l \in S_2$ induces four tuples $t_{l1}, t_{l2}, t_{l3}, t_{l4}$. For these tuples, we have $t_{lr}[V] = g(v_l) + r\delta$ for $r = 1, 2, 3, 4$. The positive value $\delta > 0$ is a constant, and all other attributes are set to 0. This group of tuples $\forall t_{lr}$ forms the set S_4. The set S_5 comprises eight tuples, four of which contain only 0 values, and the other four contain only G values. The number G is sufficiently large. This set of tuples is employed to prevent distance normalization. The relationships between the constants are as follows:

$$\begin{aligned} & G > (2n_E+2)^2 B, (2n_E+2)b \\ & B, b \gg d \gg e \gg \triangle, \delta \ (\triangle \neq \delta) \\ & G - \frac{d}{2} > n_E \end{aligned} \tag{1}$$

The set $\Sigma = \{\varphi, \varphi_1, \cdots, \varphi_{n_E}\}$ contains DCs:

$$\varphi : \forall s, t \in I, \neg(s[V] = t[V], s[D] \neq t[D]), \tag{2}$$

$$\varphi_i : \forall s, t \in I, \neg(s[E_i] = e, t[E_i] = e, s[V] \neq t[V]), \tag{3}$$

$$(i = 1, \cdots, n_E)$$

With the DCs, the tuple pairs (t_1^i, t_2^i) do not satisfy φ_i, and the tuples t_l^i and t_l with $t_l^i[V] = t_l[V]$ violate φ. If we employ linear regression to train the dependency models and set $\gamma = 0$, $k = 1$, $\kappa = 3$, $L(t_j^i) = G - \frac{e}{2}$ for all $t_j^i \in S_1$ and $L(t_{jr}^i) = G$ for all $t_{jr}^i \in S_3$, the providers of t_j^i and t_{jr}^i are drawn from $\{t_{j1}^i, t_{j2}^i, t_{j3}^i, t_{j4}^i\}$. For $L(t_l) = G - \frac{d}{2}$ ($\forall t_l \in S_2$) and $L(t_{lr}) = G$ ($\forall t_{lr} \in S_4$), the providers of t_l and t_{lr} are from $\{t_{l1}, t_{l2}, t_{l3}, t_{l4}\}$. For all $t \in S_5$, $L(t) = G$ and the providers are from S_5. Hence, $L(t)$ for all $t \in I$ remains fixed during the repair process.

Suppose that $C > 0$ is a constant. We now prove that there exists a minimal removal set $I_N \subset I$ such that $L(I \setminus I_N) \geq -C(G - \frac{d}{2}) + (9n_E + 5n_V + 8)G - \frac{e}{2}n_E - \frac{d}{2}n_V$ if and only if there exists a vertex cover VC of G with a size of $|VC| \leq C$.

If $VC \subset V$ is a vertex cover of G with size $|VC| \leq C$, for each $t_j^i \in S_1$, if $g^{-1}(t_1^i[V]) \in VC$, then t_2^i is removed (regardless of whether $g^{-1}(t_2^i) \in VC$ or not), otherwise t_1^i is removed. In S_2, if $v_l \in VC$, then t_l is removed. Let the set of removed tuples be denoted as I_N. After deleting I_N, no conflicts remain. This is because one of the tuples in each $(t_1^i, t_2^i) \not\models \varphi_i$ is removed. The remaining tuple t_j^i could conflict with a t_l if $t_j^i[V] = t_l[V]$, but such t_l is also removed. It is evident that each removal is necessary, ensuring that I_N is minimal. As for $L(I\backslash I_N)$, the original $L(I) = (10n_E + 5n_V + 8)G - en_E - \frac{d}{2}n_V$. The decrease in L caused by removing I_N is greater than $n_E(G - \frac{e}{2}) + C(G - \frac{d}{2})$. Therefore, $L(I \setminus I_N) \geq -C(G - \frac{d}{2}) + (9n_E + 5n_V + 8)G - \frac{e}{2}n_E - \frac{d}{2}n_V$.

If I_N is a removal set of I with $L(I \setminus I_N) \geq -C(G - \frac{d}{2}) + (9n_E + 5n_V + 8)G - \frac{e}{2}n_E - \frac{d}{2}n_V$, there are C tuples $t_l \in S_2$ and n_E tuples $t_j^i \in S_1$ removed. This is because, in addition to the L decrease of $n_E(G - \frac{e}{2})$ caused by removing $t_j^i \in S_1$, the decrease of $C(G - \frac{d}{2})$ may be attributed to the removal of $t_l \in S_2$ or $t_j^i \in S_1$. Suppose that there are x t_l and $y + n_E$ t_j^i deleted, where

$$C - x = y\frac{G - \frac{e}{2}}{G - \frac{d}{2}} \tag{4}$$

The left-hand side of (4) is integral. If the equation holds, the right-hand side must also be integral. Since $\frac{G-\frac{e}{2}}{G-\frac{d}{2}}$ is fractional, to ensure the left-hand side is integral, y must be an integer multiple of $G - \frac{d}{2}$. However, given that $y \leq n_E < G - \frac{d}{2}$, the only scenario in which (4) can be satisfied is when $y = 0$ and

$x = C$. Thus, the decrease of $C(G - \frac{d}{2})$ can only be attributed to the removal of C tuples $t_l \in S_2$. Suppose that VC is the set of v_l for $t_l \in I_N \cap S_2$. The size of VC is equal to the number of t_l removed, which is less than C. If VC is not a vertex cover of G, there exists an edge $e_i = (v_{l_1}, v_{l_2})$ that is not covered. So, both t_{l_1} and t_{l_2} would remain in S_2. While one of $t_1^i, t_2^i \in S_1$ remains ($g^{-1}(t_1^i[V]) = v_{l_1}, g^{-1}(t_2^i[V]) = v_{l_2}$), it would conflict with either t_{l_1} or t_{l_2}, which contradicts the non-conflict condition of $I \setminus I_N$. Therefore, VC must be a vertex cover with $|VC| \leq C$.

□

Proof of Proposition 1. For a $t_l \in I$, suppose that $S_l = |\{t_r | L(t_i, t_r) > L(t_i, t_l), t_r \in I \backslash I_C\}|$. Because $t_l \not\models \overline{M}(t_i)$, $S_l \geq k$. The smallest rank of t_l among $L(t_i, t_r)$ providers is $S_l + 1 > k$. Therefore, t_l can never provide $L(t_i, t_l)$ for t_i. □

Proof of Proposition 2. Constraint (12) ensures that at most one tuple remains for each pair $(t_i, t_l) \not\models \Sigma$ that does not satisfy Σ, thereby ensuring that the instance is non-conflict. The objective function (18), in conjunction with constraints (13) to (15), guarantees that $L(I \backslash I_N^*)$ is maximized. Regarding minimality, suppose that I_N^* is not a minimal removal set. Then there exists a tuple $t_i \in I_N^*$ such that for all $t_l \in I \setminus I_N^*$, the pair $(t_i, t_l) \models \Sigma$. For any $t_l \in I \setminus I_N^*$, let its conformance in $I \setminus I_N^*$ and $I \setminus (I_N^* \setminus t_i)$ be denoted as $L(t_l | I \setminus I_N^*)$ and $L(t_l | I \setminus (I_N^* \setminus t_i))$, respectively. The following relationship holds:

$$L(t_l | I \backslash I_N^*) \leq L(t_l | I \backslash (I_N^* \backslash \{t_i\})). \tag{5}$$

Besides, $L(t_i | I \backslash (I_N^* \backslash \{t_i\})) > 0$. So putting back t_i into $I \backslash I_N^*$ makes

$$L(I \backslash I_N^*) \leq L(I \backslash (I_N^* \backslash \{t_i\})), \tag{6}$$

contradicting $L(I \backslash I_N^*)$ being maximum. □

Proof of Proposition 3 . Proof by Contradiction: Suppose that when X is identified as the solution to the linear program (LP), there exist $X_P, X_N \subset X$ such that transforming X to X^+ and X^- results in no $x_l \in X$ satisfying the inequality $kx_l - \sum_{r=1}^{s_l - 1} y_{lr} > x_{s_l}$. The set X can be expressed as the convex combination $X = 0.5X^+ + 0.5X^-$, indicating that it is not an extreme point of the feasible region. This contradicts the assertion that X is the solution returned by the LP. If no X_N, X_P exists, X cannot be represented as any convex combination of other feasible solutions X^+, X^-. So X is an extreme point.

Note: When altering the values of x_i to $x_i + \varepsilon$ and x_l to $x_l - \varepsilon$ (where $x_i \in X_P$ and $x_l \in X_N$), the values of y_{ir} and y_{rl} are correspondingly adjusted simultaneously. If $y_{ir} = x_i$, then y_{ir} undergoes the same modification as x_i, and similarly for the case where $y_{ir} = x_r$, $y_{lr} = x_l$. The y_{lr} terms in the inequality $kx_l - \sum_{r=1}^{s_l - 1} y_{lr} > x_{s_l}$ represent the values after these modifications. □

Proof of Proposition 4. For condition (1), if for all $X_N, X_P \subset X$, either X^+ or X^- violates constraint (12), then X cannot be expressed as a convex combination of any feasible solutions. Consequently, $x_i \in X$ constitutes an extreme point of the feasible region defined by (12), which is incongruent with the definition of a $\mathcal{Y}$-solution.

For condition (2), if (20) does not hold, then X^+ is a feasible solution that yields a higher objective value than X. This contradicts the premise that X is the optimal solution. □

Proof of Proposition 5. X is an $\mathcal{X}$-solution, which implies that $x_i \in X$ constitutes an extreme point of the feasible region defined by (12). If we consider each conflict tuple t_i as a vertex and each conflict pair (t_i, t_l) as an edge, then (12) defines the constraints for the minimum vertex cover problem. As stated in [2], the feasible region of (12) is half-integral, meaning that for all $x_i \in X$, $x_i \in \{0, 0.5, 1\}^n$. Suppose there exists a y_{il} not in $\{0, 0.5, 1\}^n$. Then, there exists $x_i > y_{il}$ and $x_l > y_{il}$. By turning $y_{il} \notin \{0, 0.5, 1\}$ with $max_{t_l \in \overline{M}(t_i)} L(t_i, t_l)$ into $min\{x_i, x_l\}$ could get a solution with a larger objective. Therefore, a solution with y_{il} not in $\{0, 0.5, 1\}$ isn't optimal. □

Proof of Proposition 6. Suppose that the dirty instance I corresponds to a solution X where $x_i = 1$ for all $t_i \in I_q$ before repairing. By altering all the x_i from 1 to 0.5, the solution X becomes feasible, as for all pairs (t_i, t_l) that do not satisfy Σ, the sum $x_i + x_l = 1$.

First, we demonstrate that for all $x_i \in X$, the assignment $x_i = 0.5$ constitutes an extreme point of the feasible region (P). Consider $X = tX_1 + (1-t)X_2$, where $0 \leq t \leq 1$, and $X_1 = [x_1^1, x_2^1, ...]$, $X_2 = [x_1^2, x_2^2, ...]$. We have $x_i = tx_i^1 + (1-t)x_i^2 = 0.5$. Assume that $x_1^1 > 0.5$, then it follows that $x_i^1 < 0.5$ for all $i \neq 1$. In X_2, if $x_2^2 > 0.5$, then $x_i^2 < 0.5$ for all $i \neq 2$. For all $l \neq 1, 2$, let $x_l^1 \in X_1$ and $x_l^2 \in X_2$, both x_l^1 and x_l^2 are less than 0.5. Consequently, $tx_l^1 + (1-t)x_l^2 < 0.5$, implying that X cannot be represented as a convex combination of X_1 and X_2 with $1 > t > 0$. Therefore, $x_i = 0.5$ for all $x_i \in X$ is indeed an extreme point.

Within a clique I_q, if there is more than one $x_i \in X$ with $x_i > 0$, the only feasible solution is to set $\forall x_i \in X, x_i = 0.5$. When there is only one positive x_i in the clique, if the condition $0.5 \sum_{t_i \in I_q} L(t_i) > \max_{t_i \in I_q} L(t_i)$ is met, the fractional solution yields a higher objective value and should thus be returned. □

Proof of Proposition 7. Upon transitioning x_i^F from 1 to 0.5 for $t_i \in I_q$, the associated y_{il}^F values are also adjusted from 1 to 0.5 to meet constraint (13). Consequently, the adjusted utility $\overline{L}(t_i)$ is computed as $\sum_{t_l \in \overline{M}(t_i)} L(t_i, t_l) y_{il}^F = 0.5 L(t_i)$. In contrast, for the solution X^I, consider the scenario where t_i remains in I_q. Although $\sum_{t_l \in \overline{M}(t_i)} y_{il} = k$, the top-$k$ values of $L(t_i, t_l)$ within $I \backslash (I_q \backslash \{t_i\})$ are lower than those in the original set I, as some of the providers of the original $L(t_i, t_l)$ may have been excluded. Therefore, $\overline{L}(t_i) = \sum_{t_l \in \overline{M}(t_i)} L(t_i, t_l) y_{il}^I \leq \sum_{t_l \in M(t_i)} L(t_i, t_l) = L(t_i)$. □

Proof of Proposition 8. Because in an integral feasible solution X, at most one tuple is left in each I_q, indicating $\sum_{t_i \in I_q} x_i \leq 1$, the clique constraints added into the ILP are redundant constraints. Therefore the solution space of ILP is not impacted. □

Proof of Proposition 9. Once all clique constraints for each maximal clique are incorporated and the LP yields an $\mathcal{X}$-solution, we are concerned only with the feasible region of the relaxed problem (P').

$$(P'){:}\ x_i + x_l + u_{il} = 1. \qquad \sum_{t_r \in I_q} x_r + u_{I_q} = 1 \tag{7}$$

Here, u_{il} and u_{I_q} serve as variables for constraint standardization. The constraints $x_i + x_l + u_{il} = 1$ are introduced for conflict pairs (t_i, t_l) where t_i and t_l do not share any common cliques. The solution that includes u_{il}, u_{I_q} is represented as $X' = [x_i, u_{il}, u_{I_q}]$. All constraints can be expressed in an alternative form:

$$AX'^T = \mathbf{1}. \tag{8}$$

In this equation, A denotes the parameter matrix, and $\mathbf{1}$ is a vector of all ones. The column vectors A_i, A_{il}, and A_{I_q} correspond to the variables x_i, u_{il}, and u_{I_q}, respectively. According to [3], if X (or X') is an extreme point of (P) (or (P')), the column vectors of the positive variables in X' are linearly independent. Consequently, for any extreme point X' of (P'), the number of positive variables must be less than the number of constraints in (P'). In a chain, there is at least one positive variable (either x_i, x_l, or u_{il}) in each constraint. If an equation $x_i + x_l + u_{il} = 1$ contains two positive variables, their count exceeds the number of constraints. Similarly, within a clique constraint $\sum_{t_i \in I_q} x_i + u_{I_q} = 1$, if there are more than one positive variable, their number will also exceed the number of constraints. □

Proof of Proposition 10. A clique may be identified with $x_i = 0.5$ when there exists a clique $I_q = \{t_1, t_2, t_3\}$ of size 3 that has not been limited by a clique constraint. In the worst case, only one such size-3 clique can be detected per iteration. Following C_c^3 iterations, all potential size-three cliques will be encompassed by at least one clique constraint, so the loop terminates. □

Proof of Proposition 11. The minimal condition is ensured by Lines 16 to 18 in Alg.1, as any redundantly removed tuples are put back to $I \backslash I_N$. The time complexity of $O(mn^2)$ accounts for error detection and the computation of $L(t_i, t_l)$, which can be performed offline. The complexity of $O((Kn + c)^{3.5})$ corresponds to the execution time of the LP solver. The time complexity of $O(c^3)$ is associated with the convergence process. The time required for checking minimality is $O(c\log(c) + c^2)$. Therefore, the overall complexity is $O(mn^2 + (nK + c)^{3.5}c^3)$. □

Proof of Proposition 12. The size of $I \backslash I_N$ compared to $I \backslash I_N^*$ is $\frac{|I \backslash I_N|}{|I \backslash I_N^*|} \geq \frac{|I \backslash I_C|}{|I|}$. For each $t_i \in I \backslash I_N$, $\frac{L(t_i | I \backslash I_N)}{L(t_i | I \backslash I_N^*)} \geq \frac{k \min L(t_i, t_l)}{k \max L(t_i, t_l)} = \eta$. So the error bound of Alg.1 is $\frac{L(I \backslash I_N)}{L(I \backslash I_N^*)} \geq \eta \frac{|I \backslash I_C|}{n}$. □

Proof of Proposition 13. The minimal condition is safeguarded by Line 5 to Line 7 in Alg.2. The computational time for error detection and the evaluation of $L(t_i, t_l)$ is $O(mn^2)$. The time complexity for tuple removal is $O(n^2)$, while the complexity for the minimality check is $O(c\log(c) + c^2)$. So the overall complexity of the algorithm is dominated by the error detection and $L(t_i, t_l)$ calculation, resulting in a total complexity of $O(mn^2)$. □

Proof of Proposition 14. The expectation of $L(I \backslash I_N)$ can be calculated as,

$$E(L(I \backslash I_N)) = \sum_{t_i \in I} P_i \sum_{t_l \in \overline{M}(t_i)} P_l P_{il}^{in} L(t_i, t_l) \tag{9}$$

$P_i = \prod_{(t_l, t_i) \not\models \Sigma} P_{il}$ and $P_l = \prod_{(t_r, t_l) \not\models \Sigma} P_{lr}$ are the remaining probability of t_i and t_l. P_{il}^{in} is the probability of t_l providing $L(t_i, t_l)$ for t_i after repair. For $t_l \in M(t_i)$, $P_{il}^{in} = 1$ because they provide top-k $L(t_i, t_l)$ at the beginning. While actually, they cannot provide $L(t_i, t_l)$ if they are removed. Such probability is controlled by P_l. To find the error bound,

$$\begin{aligned} E(L(I \backslash I_N)) &\geq \left(\frac{\eta}{1+\eta}\right)^{2V} minL(t_i, t_l) \sum_{t_i \in I} \sum_{t_l \in \overline{M}(t_i)} P_{il}^{in} \\ &\geq \left(\frac{\eta}{2}\right)^{2V} minL(t_i, t_l) nk \end{aligned} \tag{10}$$

Combining with $L(I \backslash I_N^*) \leq maxL(t_i, t_l)nk$,

$$\frac{E(L(I \backslash I_N))}{L(I \backslash I_N^*)} \geq \left(\frac{\eta}{2}\right)^{2V+1}. \tag{11}$$

□

References

[1] S. Kolahi and L. V. S. Lakshmanan, "On approximating optimum repairs for functional dependency violations," in *Database Theory - ICDT 2009, 12th International Conference, St. Petersburg, Russia, March 23-25, 2009, Proceedings*, ser. ACM International Conference Proceeding Series, R. Fagin, Ed., vol. 361. ACM, 2009, pp. 53–62. [Online]. Available: https://doi.org/10.1145/1514894.1514901

[2] V. V. Vazirani, "Approximation algorithms," 2001.

[3] S.-C. Fang and S. Puthenpura, *Linear optimization and extensions: theory and algorithms*. Prentice-Hall, Inc., 1993.