Appendix

A. Proof of Theorem 1

Theorem 1. *The* DORC *problem is* NP*-hard.*

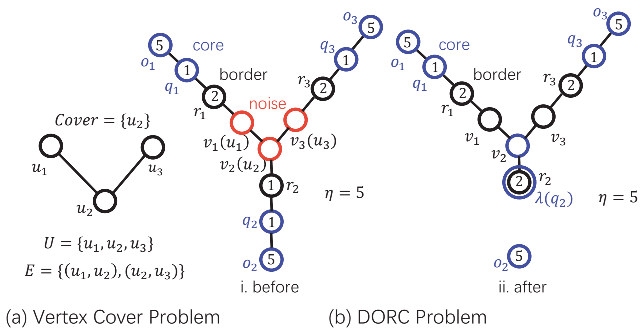

Fig. 3: DORC Transformation to Vertex Cover

Proof. We prove the conclusion by constructing a reduction from the VERTEX COVER problem, which is one of Karp's 21 NP-complete problems [18]. Given an arbitrary graph $G(U, E)$ with $|U|$ vertices and $|E|$ edges, a vertex cover is a subset $C \subseteq U$ such that for each edge $(u_i, u_j) \in E$, C contains at least one of u_i or u_j.

We next transform the aforementioned VERTEX COVER problem on $G(U, E)$ into a specific DORC problem on a set of points $\mathcal{P}$, with the following mapping from $G(U, E)$ to $\mathcal{P}$. For each edge $(u_i, u_j) \in G(U, E)$, we map it to two points $v_i, v_j \in \mathcal{P}$ with $\delta(v_i, v_j) = \varepsilon$. Moreover, for each point $v_i \in \mathcal{P}$ mapped from $u_i \in G(U, E)$, we further introduce $\eta - l - 2$ data points overlaid at a location $r_i \in \mathcal{P}$, one data point at a location $q_i \in \mathcal{P}$, and η data points overlaid at a location $o_i \in \mathcal{P}$, where l is the number of edges connected to u_i in $G(U, E)$ and $l < \eta < n$. Simultaneously, for each $v_i \in \mathcal{P}$, we define the distances between these newly introduced points in $\mathcal{P}$ as follows: $\delta(v_i, r_i) = \varepsilon$, $\delta(r_i, q_i) = \varepsilon$, and $\delta(q_i, o_i) = \varepsilon$. Note that all other pairs of points have distances $\delta(*, *) \gg \varepsilon$. This transformation can be done in polynomial time.

Note that all newly introduced points are non-noise points and all v_i are noise points. This is because points in locations o_i and q_i are all core points, with $\eta+1$ and $2\eta-l-1$ neighbors, respectively. Points in location r_i within the ε-neighborhood of core point q_i are border points. However, each v_i, having $1 + l + (\eta - l - 2) = \eta - 1$ neighbors, is identified as a noise point. With such transformation, we can prove that $G(U, E)$ has a vertex cover if and only if there is a feasible repair on $\mathcal{P}$, by the following:

If graph G has a vertex cover C of size $|C| = k$, then a repair λ modifying k points with cost $\Delta(\lambda) = k\varepsilon$ is feasible. For each $u_i \in C$ (corresponding to $v_i \in \mathcal{P}$), setting $\lambda(q_i) = r_i$ ensures that v_i acquires η neighbors and becomes a core point. By the vertex cover definition, for each $u_j \notin C$, there must be a $u_i \in C$ having an edge with u_j. The corresponding v_j falls in the ε-neighborhood of core point v_i and becomes a border point, eliminating all noise points.

Conversely, suppose that there exists a λ such that $\Delta(\lambda) = k\varepsilon$ and $k < |C^*|$, where C^* is a minimum vertex cover. As aforementioned, only all v_i are noise points. First, the repairing must happen between (v_i, v_j), (v_i, r_i), (r_i, q_i) or (q_i, o_i), given distances of other pairs $\gg \varepsilon$. Moreover, repairing between $(v_i, v_j), (v_i, v_j)$ cannot eliminate noise points v_i, since the number of neighbors of v_i will not increase. Repairing between (q_i, o_i) or from r_i to q_i cannot upgrade the corresponding r_i to core point, and thus cannot make v_i non-noise. Thus, the only feasible repairing is $\lambda(q_i) = r_i$. This repair makes v_i a core point and makes all v_j with $\delta(v_i, v_j) \le \varepsilon$, i.e., $(u_i, u_j) \in E$, become border points. Since we need to repair at least $|C^*|$ points to cover all the edges with cost $|C^*|\varepsilon$, it contradicts any claim that fewer repairs can achieve the same result. □

B. Proof of Proposition 2

Proposition 2. *For $\eta = 2$, there is a* PTIME *algorithm for solving the* DORC *problem.*

Proof. In the case of $\eta = 2$, a point can only either be a core (with at least two ε-neighbors) or a noise (with itself as the only ε-neighbor). By repairing a noise point p_i to any point $\lambda(p_i) = p_j$, both $\lambda(p_i)$ and p_j upgrade to core points.

In the PTIME algorithm for the minimum EDGE COVER problem [12], we interpret the relationships of noises with other points. The algorithm greedily picks a noise p_i with $\min_{1 \le i \le n, 1 \le j \le n, i \ne j} w(p_i, p_j)$ to repair in each step, and forms an optimal solution when no noise points remain. □

C. Proof of Proposition 3

Proposition 3. *The optimal solution $\mathbf{x}^{\text{ILP}}, \mathbf{y}^{\text{ILP}}$ of* ILP *forms an optimal repair λ^{ILP} with the minimum repairing cost*

$$\Delta(\lambda^{\text{ILP}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_{ij}^{\text{ILP}},$$

where $\lambda^{\text{ILP}}(p_i) = p_j$ iff $x_{ij}^{\text{ILP}} = 1, 1 \le i \le n, 1 \le j \le n$.

Proof. The correctness is obvious given that a point can only be repaired to one location (Formula 2) with cost weight $w_{ij} = w(p_i, p_j)$, and all the repaired points are either cores or in ε-neighborhood of some cores (Formulas 4 and 5). □

D. Proof of Proposition 4

Proposition 4. *For $\eta < n$, a feasible solution to the* ILP *problem always exists.*

Proof. By simply repairing all the points to a single location say p_1, we have $x_{i1} = 1$ and $y_1 = 1$, i.e., all the points become core points locating in p_1 after repairing. □

E. Proof of Proposition 5

Proposition 5. *When selecting noise point p_i to repair p_j,i.e.,$|\mathcal{N} \setminus \mathcal{N}(p_j)| \ge (1 - y_j)\eta$, noise point p_i does not need to be repaired from location p_i to location p_j if the distance between them is less than ε, i.e., $\delta(p_i, p_j) \le \varepsilon$.*

Proof. Supposing that noise point p_i is repaired into location p_j, $C(p_j)$ would not change, since it already has $p_i \in C(p_j)$. And the count of ε-neighbors of p_j, i.e., $|C(p_j)|$ would not

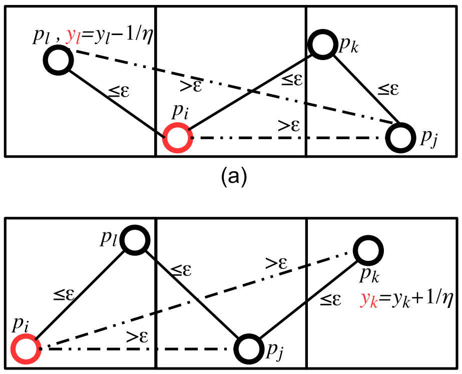

Fig.7: Cases for updating y_l and y_k for points p_l and p_k

increase. Repairing p_i into p_j will only increase the repair cost with $\delta(p_i, p_j)$. Referring to the minimum change principle in data repairing, this repair is unnecessary. Thus, there is no need to move noise point p_i into point p_j when the distance between them is less than ε. □

F. Proof of Proposition 6

Proposition 6. *Consider repairing point p_i into another point p_j, i.e., $\lambda(p_i) = p_j$. Let $\mathcal{N}_0$ denote the noise set before repair, and let $\mathcal{N}$ denote the noise set after repair. We have $\mathcal{N} \subset \mathcal{N}_0$.*

Proof. Referring to Algorithm 2, ε-neighbors of p_i and p_j, i.e., points $p_l \in C(p_i)$ and points $p_k \in C(p_j)$, will be influenced when repairing p_i into p_j.

For $p_l \in C(p_i)$, consider the distance between p_l and p_j. If $\delta(p_l, p_j) > \varepsilon$, as shown in Figure 7 (a), y_l will decrease to $y_l = y_l - \frac{1}{\eta}$ after repair, because p_i is removed from p_l's ε-neighborhood. Otherwise, if $\delta(p_l, p_j) \le \varepsilon$, as shown in Figure 7 (b), y_l will not change after repair, because p_i is still in p_l's ε-neighborhood. p_l can either be a border point or a noise point before repairing, since there is no need to repair p_i if p_l is a core point. With y_l stays the same or decrease, p_l's point type remains unchanged.

For $p_k \in C(p_j)$, consider the distance between p_k and p_i. If $\delta(p_k, p_i) > \varepsilon$, as shown in Figure 7 (b), y_k will increase to $y_k = y_k + \frac{1}{\eta}$ after repair, because p_i becomes p_k's new ε-neighbor. If $\delta(p_k, p_i) \le \varepsilon$, as shown in Figure 7 (a), y_k will not change after repair, because p_i is still in p_k's ε-neighborhood. There are three possible point types for p_k before repairing: core, border, or noise. If p_k is a core, it stays as a core. If p_k is a border, it either stays as a border or becomes a new core. If p_k is a noise, it either stays as as noise or becomes a border. This is because the number of ε-neighbors of p_k can only increase after repair.

In summary, after each repair, p_i either becomes a core or border point by moving into other locations, p_j becomes a core or stays as a border with y_j increased, $p_l \in C(p_i)$ stays as before, and $p_k \in C(p_j)$ has either the same y_k or an increased y_k. Thus, no new noise will be introduced. □

G. Proof of Proposition 7

Proposition 7. *The time complexity of the grid-based* GDORC *method is* $\mathrm{O}(nN_m + N\eta n)$*, with* $\mathrm{O}(nN_m)$ *being the time complexity of* Algorithm 1 (INITIALIZATION) *and* $\mathrm{O}(N\eta n)$ *being the time complexity of* Algorithm 2 (GDORC REPAIR).

Proof. Algorithm 1 first traverses each point on the cell level, with Line 1 going over each cell, and Line 3 going over each point in each cell, costing $\mathrm{O}(n)$ for n points. Then, it calculates each point's $\varepsilon-$neighbor cell in Line 8. Given cell width $\frac{\varepsilon}{\sqrt{d}}$, there are $\mathrm{O}((2\lceil\sqrt{d}\rceil + 1)^d)$ cells to check for each cell's $\varepsilon-$neighbor. After that, Line 9 checks for points in these neighbors, with each cell containing at most N_m points. In sum, we need to check $\mathrm{O}((2\lceil\sqrt{d}\rceil + 1)^d N_m)$ points to determine the status of each point. Since the dimensionality d is low, $\mathrm{O}((2\lceil\sqrt{d}\rceil + 1)^d)$ can be regarded as constant. Therefore, initialization on cell level takes $\mathrm{O}(nN_m)$ time. Similarly, initialization on point level also takes $\mathrm{O}(nN_m)$ time. To initialize supporting sets, every point is traversed again, with $\mathrm{O}(n)$ time. In summary, the time complexity of Algorithm 1 is $\mathrm{O}(nN_m)$.

For repairing in Algorithm 2 (GDORC), Line 2 for finding point $p_j \in \mathcal{P}$ with the maximum $y_j < 1$ can be solved in $\mathcal{O}(1)$ time, by amortizing points $\mathcal{P}$ into a constant space w.r.t. $\mathbf{y}$ values. After selecting a p_j to repair, there are at most η points to repair. Lines 4-12 will repeat $\mathrm{O}(\eta)$ times. When there are sufficient noises, it takes $\mathrm{O}(N_s)$ to find the nearest noise point with the minimum cell distance $\delta(u_j, u_i)$ in Line 5 every time. It costs $\mathrm{O}(1)$ to repair the selected p_i in Line 7 and update set status in Lines 8-11. In sum, when there are sufficient noise point for repairing, it takes $\mathrm{O}(\eta N_s)$ to repair a p_j. With $N_s < N$, the time cost can be reduced to $\mathrm{O}(\eta N)$.

When there are insufficient noises left to repair p_j into a core, Lines 17-20 allocate the remaining noises. For each remaining noise, it takes $\mathrm{O}(N_c)$ to find the nearest core with the minimum cell distance $\delta(u_j, u_i)$ in Line 19. With at most η noises remained, it takes $\mathrm{O}(\eta N_c)$ to repair all the remaining noises. With $N_c < N$, the time cost can be reduced to $\mathrm{O}(\eta N)$.

In the worst case, every border point needs only one noise point to repair, so Line 1 runs $|\mathcal{N}|$ times. Since $|\mathcal{N}| << n$, Algorithm 2 costs $\mathrm{O}(N\eta n)$. □

H. Proof of Proposition 8

Proposition 8. Algorithm 2 (GDORC REPAIR) *returns a feasible solution to the* ILP *problem, and is a factor-$\alpha\eta$ approximation with $\alpha = \frac{\delta_{\max}}{\delta_{\min}}$.*

Proof. The correctness of the grid-based method is verified by Proposition 6 since Algorithm 2 (GDORC) can repair all noises without introducing new noises.

The repairing cost every time is no greater than $\delta_{\max}$ since Algorithm 2 repairs only the points in $\mathcal{N}$, i.e.,

$$\Delta(\lambda^{\text{GDORC}}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij} \le \delta_{\max}|\mathcal{N}|. \quad (8)$$

Furthermore, a noise point $p_i \in \mathcal{N}$ can be addressed with repairing p_i itself and its ε-neighbor p_j into a core point via repairing (with cost at least $\delta_{\min}$). When repairing p_i or p_j

into a core point, it eliminates at most η noises. Considering all the $|\mathcal{N}|$ noise points, there are at least $\frac{|\mathcal{N}|}{\eta}$ points repaired, with cost no less than $\frac{|\mathcal{N}|}{\eta}\delta_{\min}$, i.e.,

$$\Delta(\lambda^{\text{ILP}}) \geq \frac{|\mathcal{N}|}{\eta}\delta_{\min}. \tag{9}$$

With formulas (8) and (9), we can derive that

$$\frac{\Delta(\lambda^{\text{GDORC}})}{\Delta(\lambda^{\text{ILP}})} \leq \frac{\delta_{\max}}{\delta_{\min}}\eta,$$

it concludes the factor-$\alpha\eta$ approximation. □

I. Proof of Proposition 9

Proposition 9. *For $\eta = 2$, Algorithm 2 (*GDORC*) is a factor-α approximation with $\alpha = \frac{\delta_{\max}}{\delta_{\min}}$.*

Proof. With the proof of Proposition 2, a point is either a core point (with $y_j^{\text{LP}} = 1$) or a noise point (with $y_j^{\text{LP}} = \frac{1}{2}$) in this special case. That means Line 2 in Algorithm 2 always selects a noise point in $\mathcal{N}$. It was repaired with another $p_i \in \mathcal{N}$ in Line 5 to Line 7 or moved to another $p_i \in \mathcal{P}_c$ in Line 18 to Line 20, where $\mathcal{P}_c < \mathcal{P} \setminus \mathcal{N}$ We have $\Delta(\lambda^{\text{GDORC}}) \leq (\delta_{\max})\frac{|\mathcal{N}|}{2}$. Combining with Formula 9, where $\eta = 2$, it follows $\frac{\Delta(\lambda^{\text{GDORC}})}{\Delta(\lambda^{\text{ILP}})} \leq \alpha$. □