APPENDIX: OMITTED PROOFS

Proof of Lemma 1. Firstly, it is easy to see that for any two vertices in the same connected component, their corresponding core objects are density-reachable from each other and thus belong to the same cluster. Secondly, for any two core objects in the same cluster, they must be density-reachable from each other, and thus belong to the same connected component of G_ε. Hence, the lemma holds. □

Proof of Lemma 2. Firstly, for any object, if it is a core object w.r.t. ε_1, then it is also a core object w.r.t. ε_2 that is larger than ε_1; thus, $V(G_{\varepsilon_1}) \subseteq V(G_{\varepsilon_2})$. Secondly, it is obvious that $E(G_{\varepsilon_1}) \subseteq E(G_{\varepsilon_2})$ based on the fact that $\varepsilon_1 < \varepsilon_2$. Consequently, the lemma holds. □

Proof of Lemma 3. For any ε, G_ε consists of exactly the edges of G_c with weight no larger than ε as discussed above, and also some potentially isolated vertices. Then, following the property of the minimum spanning forest in an edge-weighted graph, the connected components of the subgraph of F_c that consists of all edges of weights no larger than ε are the same as the non-singleton connected components of G_ε. Hence, the lemma follows from Lemma 1. □

Proof of Lemma 4. Let's consider an arbitrary $\varepsilon' > \varepsilon$. Obviously, o' is still a core object w.r.t. ε'. For ε', if o is still a non-core object, then o is a border object and belongs to the same cluster as o' because of $d(o, o') \leq \varepsilon < \varepsilon'$. Otherwise, o becomes a core object w.r.t. ε'; then o and o' also belong to the same cluster since $d(o, o') \leq \varepsilon < \varepsilon'$. □

Proof of Theorem 1. Following the discussions in Section 4.1, it is easy to see that the core object clusters are correctly obtained. We prove in the following that the border objects are correctly assigned to clusters. Note that, it is easy to see that all the assignments of border objects to clusters in Lines 18–19 are correct. Let's consider a border object o w.r.t. the query ε. Firstly, if there exists a triplet $(o, o^*, w_b(o, o^*)) \in B$ such that $w_b(o, o^*) \leq \varepsilon$. Then, o^* is a core object w.r.t. ε, and o is assigned to o^*'s cluster in Lines 18–19. Secondly, if o belongs to a cluster in the density-based clustering for ε, then there must exist a core object o' (i.e., $o'.C \leq \varepsilon$) from which o is directly density-reachable (i.e., $d(o, o') \leq \varepsilon$); as a result, there must exists a triplet $(o, o^*, w_b(o, o^*)) \in B$ with $w_b(o, o^*) \leq \varepsilon$.

Regarding the time complexity, we use a forest of parent pointer trees to represent the disjoint-set data structure [11], and the representative object in a set is the one at the root of the tree. Let $\|\mathcal{K}\|$ denote the total number of distinct vertices in the clusters of $\mathcal{K}$. Then, both the number of processed edges of F_c (by Line 3) and the number of processed triplets of B (by Line 12) are bounded by $\|\mathcal{K}\|$. By adopting both the path compression optimization and the union by rank optimization in the disjoint-set data structure, the total time complexity of Lines 3–7 is $O(\|\mathcal{K}\| \cdot \alpha(\|\mathcal{K}\|))$, where $\alpha(\cdot)$ is the inverse Ackermann function. As $\alpha(\cdot)$ is an extremely slow-growing function, we omit this term in our time complexity analysis. By adopting the path compression optimization, Lines 9–19 take time $O(\|\mathcal{K}\|)$. Thus, the time complexity follows. □

Proof of Theorem 2. Firstly, computing $o.C$ for all objects in D (i.e., Lines 1–3) takes $O(T_{nei} + m \log \mu)$ time. Specifically, Lines 1–2 take time T_{nei}. For an object $o \in D$, running Line 3 takes $O(|N(o)| \log \mu)$ time, by using a priority queue of size μ to store the potential μ closest neighbors. Thus, running Line 3 for all objects of D takes $O(m \log \mu)$ time.

Secondly, computing the minimum spanning forest F_c (i.e., Lines 5–22) takes $O(T_{nei}+m+|D| \log |D|)$ time. Specifically, running Lines 14 for all objects takes T_{nei} time, the same as Lines 1–2. In addition, the Prim's algorithm has a time complexity of $O(m + n \log n)$ for a graph with n vertices and m edges, when using the Fibonacci heap.

Thirdly, running Lines 23–26 for all objects take $O(m)$ total time.

Thus, the time complexity holds. The space complexity is obvious as we do not materialize the core graph G_c. □

Proof of Lemma 5. Recall that ε_{o_1,o_2} is the smallest ε such that o_1 and o_2 belong to the same core object cluster. Following Lemma 3, it is the smallest ε such that o_1 and o_2 are in the same connected component of the subgraph of F_c that consists of all edges of weights no larger than ε. Consequently, it is the maximum edge weight in the unique path between o_1 and o_2 in F_c. □

Proof of Lemma 6. Recall that $D_o = \{o' \in D \mid \max\{o'.C, d(o, o')\} < o.C\}$, where $o.C$ is the μ-th smallest distance among the distances between o and its neighbors $N(o)$. Hence, at most $\mu - 1$ objects o' (including o itself) can satisfy $d(o, o') < o.C$, which implies $|D_o| < \mu - 1$. □