Appendix

for Reduction Algorithms for Persistence Diagrams of Networks: CoralTDA and PrunIT

This appendix gives the proofs of our theorems, list the pseudocode of CoralTDA and Prunit and show further reduction results in networks.

A Datasets

The following table lists characteristics of graphs in our datasets.

Table 2: Characteristics of the datasets in graph and node classification experiments.

Dataset	NumGraphs	AvgNumNodes	AvgNumEdges
DD	1178	284.32	715.66
DHFR	467	42.43	44.54
ENZYMES	600	32.6	62.14
FIRSTMM	41	1377.27	3074.10
NCI1	4110	29.87	32.30
OHSU	79	82.01	199.66
PROTEINS	1113	39.06	72.82
REDDIT-BINARY	2000	429.63	497.75
SYNNEW	300	100.0	196.25
TWITTER	973	83.5	1817
FACEBOOK	10	403.9	8823.4
CORA	1	2708	5429
CITESEER	1	3264	4536
ARXIV	169343	33	111.8
MAG	1939743	31	112.5

B Algorithms

This section gives the CoralTDA reduction (Algorithm 1) and PrunIT algorithm (Algorithm 2). Our pseudocode is optimized for clarity. Our implementation is available at github.com/cakcora/PersistentHomologyWithCoralPrunit.

Algorithm 1 CoralTDA

Input: k and $\mathcal{G}$
Output: $\mathcal{G}^{k+1}$

1: flag=true
2: $\mathcal{G}_{k+1} = \mathcal{G}$
3: **while** flag is true and $\mathcal{G}_{k+1}$ is not empty **do**
4: flag=false
5: **for** $u \in \mathcal{V}$ **do**
6: **if** $|\mathbb{N}(u)| < (k+1)$ **then**
7: flag=true
8: $\mathcal{G}_{k+1} = \mathcal{G}_{k+1} \setminus u$
9: **end if**
10: **end for**
11: **end while**
12: **return** $\mathcal{G}_{k+1}$

In Algorithm 1, we repeatedly delete vertices that have less than k+1 edges and return the resulting graph. In Algorithm 2, we search for dominated neighbors of a vertex (lines 7–14) and delete such a neighbor (if exists) from the graph. Here $\mathbb{N}(u)$ denotes the neighbors of u (excluding u). We continue

Algorithm 2 PrunIT

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: $\mathcal{G}'$

1: iterate=true
2: $\mathcal{G}' = \mathcal{G}$
3: **while** iterate is true **do**
4: iterate=false
5: **for** $u \in \mathcal{V}$ **do**
6: **for** $w \in \mathbb{N}(u)$ **do**
7: $dom_{u\to w} = dom_{w\to u} = false$
8: **for** $n \in \{\mathbb{N}(u) \cup \mathbb{N}(w)\}$ **do**
9: **if** $e_{un} \in \mathcal{E} \wedge e_{wn} \notin \mathcal{E}$ **then**
10: $dom_{u\to w} = true$
11: **end if**
12: **if** $e_{uw} \notin \mathcal{E} \wedge e_{vw} \in \mathcal{E}$ **then**
13: $dom_{w\to u} = true$
14: **end if**
15: **end for**
16: **if** $dom_{u\to w} = true \wedge dom_{w\to u} = false$ **then**
17: $\mathcal{G}' = \mathcal{G}' \setminus w$ and $iterate = true$
18: **else if** $dom_{u\to w} = false \wedge dom_{w\to u} = true$ **then**
19: $\mathcal{G}' = \mathcal{G}' \setminus u$ and $iterate = true$
20: **end if**
21: **end for**
22: **end for**
23: **end while**
24: **return** $\mathcal{G}'$

until we cannot find a dominated vertex. Lines 4–24 has a complexity of $\mathcal{O}(|V|^3)$. The outer loop (line 3) may run $|V|$, however we can parallelize the algorithm for each vertex.

C Proofs of the Theorems

C.1 Proof of CoralTDA

First, we give the proof of Theorem 2. We use the same notation introduced in Section 4.

Theorem 2: *Let $\mathcal{G}$ be an unweighted connected graph. Let $f : \mathcal{V} \to \mathbb{R}$ be a filtering function on $\mathcal{G}$. Let $PD_k(\mathcal{G}, f)$ represent the k^{th} persistence diagram for the sublevel filtration of the clique complexes. Let $\widehat{\mathcal{G}}^k$ be the k-core of $\mathcal{G}$. Then, for any $j \geq k$*

$$PD_j(\mathcal{G}, f) = PD_j(\mathcal{G}^{k+1}, f)$$

Proof: For simplicity, we prove the theorem for $j = k$. Then, we give the generalization to $j > k$ case. Fix $k \geq 1$. In order to show the theorem, we need to prove that $(b, d) \in PD_k(\widehat{\mathcal{G}})$ if and only if $(b, d) \in PD_k(\widehat{\mathcal{G}}^{k+1})$. In other words, if a k^{th}-homology class $\sigma = [S]$ is born at $\widehat{\mathcal{G}}_b$ and dies at $\widehat{\mathcal{G}}_d$, then the same homology class $[S]$ is born at $\widehat{\mathcal{G}}_b^{k+1}$ and dies at $\widehat{\mathcal{G}}_d^{k+1}$.

We first prove that the inclusion map $\widehat{\mathcal{G}}_i^{k+1} \hookrightarrow \widehat{\mathcal{G}}_i$ in the Diagram 1 induces an isomorphism for the homology groups $H_k(\widehat{\mathcal{G}}_i) = H_k(\widehat{\mathcal{G}}_i^{k+1})$ for any $0 \leq i \leq m$. Then, the proof of the theorem will follow by the equivalence of the induced persistence modules.

For simplicity, we omit the subscript i. Assume $\sigma \in H_k(\widehat{\mathcal{G}})$. In particular, σ is a k-homology class of $\widehat{\mathcal{G}}$. This means there exists a k-cycle S in the chain complex $C_k(\widehat{\mathcal{G}})$ with $\sigma = [S]$. We claim that any vertex v in S must have degree at least $k + 1$, and hence $S \in C_k(\widehat{\mathcal{G}}^{k+1})$.

As S is k-cycle $\partial_k S = 0 \in C_{k-1}(\widehat{\mathcal{G}})$. Let $S = \sum_{i=1}^{N} c_i \Delta_i$ where $\{\Delta_i\}$ are k-simplices in the simplicial complex $\widehat{\mathcal{G}}$. Then, $\partial S = \partial \sum_i c_i \Delta_i = \sum_i c_i \partial \Delta_i$. Notice that if $\Delta_i \subset \widehat{\mathcal{G}}$, this implies any

vertex of $\Delta_i = [w_0^i, w_1^i, \ldots, w_k^i]$ has at least degree k in $\mathcal{G}$ as Δ_i has $k+1$ vertices $\{w_0^i, w_1^i, \ldots, w_k^i\}$, and they are all pairwise connected by an edge in $\mathcal{G}$. Now, $\partial_k \Delta_i = \sum_{r=0}^{k} (-1)^r \Omega_r^i$ where $\Omega_r^i = [w_0^i, w_1^i, \ldots, w_{r-1}^i, w_{r+1}^i, \ldots, w_k^i]$ are $k-1$ simplices in $\widehat{\mathcal{G}}$ for $0 \leq r \leq k$, i.e. Ω_r^i is a $(k-1)$-face of Δ_i.

As S being a k-cycle, and $\partial_k S = 0$, the sum $\sum_i c_i \partial \Delta_i = 0$. This implies that in the sum any $(k-1)$-chain Ω_r^i must cancel out with another $\Omega_s^j \subset \partial \Delta_j$ where Δ_j contains Ω_r^i and a vertex w_s^j which is not a vertex in Δ_i. Therefore, any vertex in Δ_i is connected to both all other vertices $\{w_r^i\}$ in Δ_i and another vertex $w_s^j \in \Delta_j \subset S$. Hence, any vertex in Δ^i has degree $k+1$. This can be generalized to any vertex in S. Hence, any vertex in S has degree at least $k+1$. This proves by induction that $S \subset \widehat{\mathcal{G}}^{k+1}$ as follows. All vertices in S has degree $k+1$ in $\mathcal{G}$. Notice that the vertices of S has degree $\geq k+1$ because of the other vertices in S, not with the help of outsider vertices. So, as long as all the vertices in S are still in i-core $\mathcal{G}^i$, then any vertex of S will still have degree $\geq k+1$ in $\mathcal{G}^i$. Therefore, By going inductively on the core index i, none of the vertices of S are removed from $\mathcal{G}^i$ for $1 \leq i \leq k+1$. Therefore, S is a k-cycle in $\widehat{\mathcal{G}}^{k+1}$, i.e. $S \in C_k(\widehat{\mathcal{G}}^{k+1})$.

This proves for $\partial_k : C_k(\widehat{\mathcal{G}}) \to C_{k-1}(\widehat{\mathcal{G}})$ and $\partial_k^{k+1} : C_k(\widehat{\mathcal{G}}^{k+1}) \to C_{k-1}(\widehat{\mathcal{G}}^{k+1})$, we have $\ker\partial_k = \ker\partial_k^{k+1}$ (Recall $\widehat{\mathcal{G}}^{k+1} \subset \widehat{\mathcal{G}}$). Notice that any vertex in any $(k+1)$-simplex in $\widehat{\mathcal{G}}$ has degree at least $k+1$. Then, $C_{k+1}(\widehat{\mathcal{G}}) = C_{k+1}(\widehat{\mathcal{G}}^{k+1})$. This implies $\mathrm{im}\partial_{k+1} = \mathrm{im}\partial_{k+1}^{k+1}$ where $\partial_{k+1}^{k+1} : C_{k+1}(\widehat{\mathcal{G}}^{k+1}) \to C_k(\widehat{\mathcal{G}}^{k+1})$. Then, $H_k(\widehat{\mathcal{G}}) = H_k(\widehat{\mathcal{G}}^{k+1})$. This proves any k-cycle S must be produced by the k-simplices in $\widehat{\mathcal{G}}^{k+1}$, and lower degree vertices cannot belong to S.

Also, k-core of $\mathcal{G}_i$ is equal to i^{th} step of the filtration induced by $f : \mathcal{V}^k \to \mathbb{R}$, i.e.$(\mathcal{G}^k)_i = (\mathcal{G}_i)^k$. Hence, if σ is born in $H_k(\widehat{\mathcal{G}}_b)$, then it is also born in $H_k(\widehat{\mathcal{G}}_b^{k+1})$. If it dies in $H_k(\widehat{\mathcal{G}}_d)$, it also dies in $H_k(\widehat{\mathcal{G}}_d^{k+1})$. This proves $PD_k(\mathcal{G}) = PD_k(\mathcal{G}^{k+1})$.

For the generalization in $j > k$ case, one only needs to consider the same process for higher order cycles. For some $j > k$, let S be a j-cycle in $\widehat{\mathcal{G}}$, i.e. $C_j(\widehat{\mathcal{G}})$. Then, by above any vertex in S must have degree at least $j+1$. This implies S must be j-cycle in $\widehat{\mathcal{G}}^{k+1}$, too.

Similarly, any vertex in a $(j+1)$-simplex must have degree $\geq k+1$. This means $C_{j+1}(\widehat{\mathcal{G}}) = C_{j+1}(\widehat{\mathcal{G}}^{k+1})$. This implies $H_j(\widehat{\mathcal{G}}) = H_j(\widehat{\mathcal{G}}^{k+1})$ for any $j \geq k$. This proves that the inclusion $\widehat{\mathcal{G}}_i^{k+1} \hookrightarrow \widehat{\mathcal{G}}_i$ induces an isomorphism for the homology groups $H_j(\widehat{\mathcal{G}}_i) = H_j(\widehat{\mathcal{G}}_i^{k+1})$ for any $j \geq k$. This shows the equivalence of the induced persistence modules. The proof of the theorem follows. □

C.2 Proof of PrunIT

Now, we give the proof of Theorem 7. We use the same notation given in Section 5.

Theorem 7: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an unweighted graph, and $f : \mathcal{V} \to \mathbb{R}$ be a filtering function. Let $u \in \mathcal{V}$ be dominated by $v \in \mathcal{V}$ and $f(u) \geq f(v)$. Then, removing u from $\mathcal{G}$ does not change the persistence diagrams for sublevel filtration, i.e. for any $k \geq 0$

$$PD_k(\mathcal{G}, f) = PD_k(\mathcal{G} - \{u\}, f).$$

Proof: Let $\widehat{\mathcal{G}}_0 \subset \widehat{\mathcal{G}}_1 \subset \widehat{\mathcal{G}}_2 \subset \cdots \subset \widehat{\mathcal{G}}_m$ be the sequence of clique complexes in the induced sublevel filtration. Let $\alpha_{i_0-1} < f(v) \leq \alpha_{i_0}$. This means for any $j \geq i_0$, $v \in \mathcal{G}_j$, and hence v is a vertex in $\widehat{\mathcal{G}}_j$. Since $f(u) \geq f(v)$, $\alpha_{i_1-1} < f(u) \leq \alpha_{i_1}$ for some $i_1 \geq i_0$. Similarly, this implies u first appears in $\mathcal{G}_{i_1}$, and $u \in \widehat{\mathcal{G}}_j$ for $j \geq i_1$. In particular, v belongs to all $\mathcal{G}_j$ containing u.

Let $\mathcal{G}' = \mathcal{G} - \{u\}$. Define the sublevel filtration $\widehat{\mathcal{G}}'_0 \subset \widehat{\mathcal{G}}'_1 \subset \widehat{\mathcal{G}}'_2 \subset ... \subset \widehat{\mathcal{G}}'_m$ for the same filtering function (restricted to $\mathcal{V}'$) $f : \mathcal{V}' \to \mathbb{R}$ with the same threshold set $\mathcal{I}$.

For any $j < i_1$, $\widehat{\mathcal{G}}_j = \widehat{\mathcal{G}}'_j$ as $u \notin \mathcal{G}_j$. Fix $j \geq i_1$. Then, $v \in \widehat{\mathcal{G}}_j$ as $f(u) \geq f(v)$. As v dominates u in $\mathcal{G}$, v dominates u in $\mathcal{G}_j$, too. Then, $\mathcal{G}_j$ folds onto $\mathcal{G}'_j = \mathcal{G}_j - \{u\}$. Then, by Lemma 5, we have homotopy equivalence $\widehat{\mathcal{G}}_j \sim \widehat{\mathcal{G}}'_j$. Hence, for any $j \geq i_1$, this gives $\widehat{\mathcal{G}}_j \sim \widehat{\mathcal{G}}'_j$.

Recall that for any $j < i_1$, $\widehat{\mathcal{G}}_j = \widehat{\mathcal{G}}'_j$. Therefore, for any $0 \leq j \leq m$, the inclusion $\widehat{\mathcal{G}}'_j \hookrightarrow \widehat{\mathcal{G}}_j$ induces an isomorphism between the homology groups $H_k(\widehat{\mathcal{G}}'_j) \simeq H_k(\widehat{\mathcal{G}}_j)$. This proves the equivalence of the induced persistence modules, and hence the corresponding persistence diagrams, i.e. $PD_k(\mathcal{G}) = PD_k(\mathcal{G}')$ for any $k \geq 0$. The proof follows. □

Now, we prove Theorem 10. Recall that n^{th} power of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as $\mathcal{G}^n = (\mathcal{V}, \mathcal{E}_n)$ Where $\mathcal{E}_n = \{e_{uv} \mid d(u, v) \leq n\}$. In other words, $\mathcal{E}_n$ is obtained by adding new edges to $\mathcal{E}$ connecting all the vertices whose graph distance $\leq n$. Then, the power filtration of $\mathcal{G}$ is defined as $\widehat{\mathcal{G}}^0 \subset \widehat{\mathcal{G}}^1 \subset \widehat{\mathcal{G}}2 \subset \cdots \subset \widehat{\mathcal{G}}^N$ where $\widehat{\mathcal{G}}^n$ is the clique complex of $\mathcal{G}^n$, n^{th}-power of $\mathcal{G}$ and $\widehat{\mathcal{G}}^0$ represent the vertex set $\mathcal{V}$ [3].

Theorem 10: [PrunIt for Power Filtration] Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an unweighted connected graph. Let $\widehat{PD}_k(\mathcal{G})$ represent k^{th} persistence diagram of $\mathcal{G}$ with power filtration. Let $u \in \mathcal{V}$ be dominated by $v \in \mathcal{V}$. Then, for any $k \geq 1$,

$$\widehat{PD}_k(\mathcal{G}) = \widehat{PD}_k(\mathcal{G} - \{u\}).$$

Proof: Notice that power filtration for a connected graph is trivial in dimension 0 as all features but one dies at threshold 1. So we will assume $k \geq 1$. Let $\mathcal{H} = \mathcal{G} - \{u\}$. We will show that $\widehat{\mathcal{G}}^n$ is homotopy equivalent to $\widehat{H}^n$ for $n \geq 1$.

Let $\{v = w_0, w_1, w_2, \ldots, w_m\}$ be all adjacent vertices to u in $\mathcal{G}$. As u is dominated by v, $\{w_1, w_2, \ldots, w_m\}$ will be adjacent to v, too. We claim that for any vertex $z \neq u$ in $\mathcal{G}$, $d(z, v) \leq d(z, u)$ where $d(z, z')$ is the length of the shortest path from z to z' and each edge has length 1. In particular, any shortest path γ from z to u must go through one of the adjacent vertices $\{v, w_1, w_2, \ldots, w_m\}$. Let $\tau \subset \gamma$ be the segment starting at z and ending with one of these adjacent vertices, say w_i. Then, $d(z, u) = \|\gamma\| = \|\tau\| + 1$. Now, consider $d(z, v)$. Since $\gamma' = \tau \cup [w_i, v]$ is a path from z to v, this implies $d(z, v) \leq \|\tau\| + 1 = d(z, u)$.

Now, we claim that if u is dominated by v in $\mathcal{G}$, then u is dominated by v in $\mathcal{G}^n$ for any n. All we need to show that any adjacent vertex to u in $\mathcal{G}^n$ is also adjacent to v. If z is adjacent to u in $\mathcal{G}^n$, then $d(z, u) \leq n$ in $\mathcal{G}$ by the definition of $\mathcal{G}^n$. By above, we have $d(z, v) \leq d(z, u) \leq n$. This implies z is adjacent to v, too. Hence, u is dominated by v. Furthermore, as v dominates u in $\mathcal{G}$, no shortest path goes through u unless the endpoint is u. Therefore, the distances in $\mathcal{G}$ and $\mathcal{H}$ would be same for any two vertices $z, z' \in \mathcal{V} - \{u\}$. Then, by Lemma 5, $\widehat{\mathcal{G}}^n$ homotopy equivalent to $\widehat{\mathcal{H}}^n$ for any $n \geq 1$.

Then, consider $\widehat{PD}_k(\mathcal{G})$ for $k \geq 1$. For $k \geq 1$, any k-dimensional topological feature will be born in $\widehat{\mathcal{G}}^n$ where $n \geq 1$. As $\widehat{\mathcal{G}}^n$ homotopy equivalent to $\widehat{\mathcal{H}}^n$ for any $n \geq 1$, the proof follows. □

Remark 11. [CoralTDA for Power Filtration] Of course, after the previous result, the natural follow-up question is *"Does CoralTDA also extends to power filtrations?"*. Unfortunately, the answer to this question is "No". The reason is by the nature of power filtration, one keeps adding edges to the existing vertices, and the degrees of the vertices increase in each step. This means k-cores significantly changes during this process. A simple counterexample can be found in [1, Corollary 6.7], where the author completely classifies the topological types of clique complexes of the powers of cyclic graphs $\{C_n\}$, i.e., A graph which forms a circle. e.g., C_5 is a connected graph with 5 vertex and 5 edges. By definition, any 3-core of a cyclic graph is empty. However, Adamazsek's result show that for any dimension k, there exists topological features $\widehat{PD}_k(C_n)$ is nontrivial for $n \geq 2k + 3$. If coralTDA could be extended to power filtrations, then $\widehat{PD}_k(C_n)$ would be trivial for any $k \geq 3$.

Remark 12. [Combining Reduction Algorithms with Graph Filtration Learning] In [29], the authors studied a fundamental question, which is to find the "best filtering function" for the given classification problem. Indeed, the "right" filtering function is often the key behind the classification performance. Under the assumption that dominated vertex value is greater than its dominating vertex value, we can invoke our PrunIt result and run the "Graph Filtration Learning" algorithm of [29] on the pruned graph which will lead to computational and (possibly) performance gains. For node classification tasks, it would be also wise to check the relation between the class differences between dominated and their dominating vertices. With similar reasoning, our CoralTDA method can also be combined with the "Graph Filtration Learning" algorithm and other methods.

Remark 13. [PrunIT vs. Strong Collapse] In [7, 8, 9], the authors introduced Strong Collapse method to effectively reduce the computational costs for the persistence diagrams. In this context, Strong Collapse method is defined for any simplicial complex sequence, whereas PrunIT ise specifically defined for the graph setting. The main difference between Strong Collapse and PrunIt is that PrunIT captures the dominated vertices in the graph at once, before filtration while Strong Collapse needs to be applied in each filtration step. In particular, for a given graph $\mathcal{G}$ and the filtering function f, PrunIT detects the dominated vertices in the graph, and by removing them, gives a smaller graph $\mathcal{G}'$ with the same persistence diagrams $PD(\mathcal{G}, f) = PD(\mathcal{G}'f)$. Since the detection of dominated vertices is done in the graph itself, the PrunIt algorithm works very fast. On the other hand, Strong Collapse method is designed to work with flag complexes. Therefore, if one wants to apply Strong Collapse method to compute $PD(\mathcal{G}, f)$, one needs to go to the filtration sequence (flag complexes) $\widehat{\mathcal{G}}_1 \subset \widehat{\mathcal{G}}_2 \subset \cdots \subset \widehat{\mathcal{G}}_N$, and apply the Strong Collapse method for each flag complex $\widehat{\mathcal{G}}_n$ for $1 \leq n \leq N$ in the the filtration sequence. This means one needs to detect dominated vertices for each $\widehat{\mathcal{G}}_n$ and remove them. Therefore, especially if the filtration sequence is long (N is large), PrunIT would be a more efficient method in graph setting.

We run experiments on the on Email-Enron dataset with 36K nodes and 183K edges (see Table 1) for PrunIt and Strong Collapse methods with degree filtering function. We provide the results in the supplementary material. In particular, we tried two different threshold step sizes $\delta_1 = 4$ and $\delta_2 = 12$ which give filtration sequence of lengths $N_1 = 346$ and $N_2 = 116$ respectively. In Table 3, we give the computation times in seconds for PrunIT and Strong Collapse algorithm to eliminate dominated vertices before PH computations. For PH computation step, recall that PH computation is cubic in terms of count of simplices [48]. In Table 3, we also give these simplex counts for each method. The details can be found in the supplementary material.

Table 3: Comparison results for PrunIt and Strong Collapse Method for Email-Enron dataset.

	Computation time (sec)		Simplex Count (million)	
Step size	**PrunIt**	**S. Collapse**	**PrunIT**	**S.Collapse**
4	1412	7014	270.2	465.2
12	513	2520	90.7	155.8

D Further Reduction Results

D.1 Reduction for Graph and Node Classification Datasets

In the following Figures 7 to 9, we report further reduction percentages for each $PD_k(\mathcal{G})$ for the graph and node classification datasets given in Table 2. Here, the x-axes represent the dimension k for $PD_k(\mathcal{G})$. Edge and simplex reduction figures closely resemble the vertex reduction in Figure 4. In time reduction of Figure 8, we see three datasets (OHSU, FACEBOOK, TWITTER) with diverging behavior as follows.

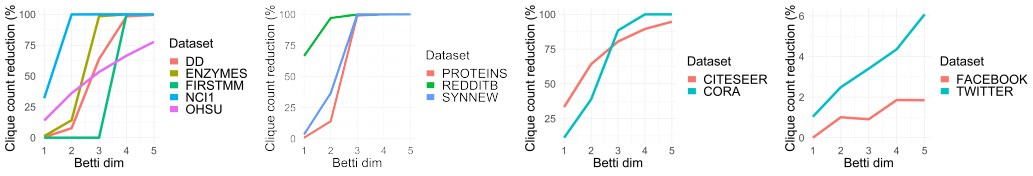

Figure 7: CoralTDA clique count reduction in graph and node classification datasets (higher is better).

OHSU, despite losing many of vertices in core reduction, does not have a time reduction as much as the other kernel datasets. We explain this behavior with small graph orders but high coreness in OHSU. Note that unlike any other kernel dataset, vertex reduction does not reach 100% in OHSU graphs. Calling core decomposition on OHSU graphs adds a time cost that takes away the benefit of coralTDA to some extent. As a result, time reduction is at most 25%.

A similar dynamic is seen in TWITTER and FACEBOOK, where graphs have high cores. As a result, most vertices cannot be peeled away in core reduction (Figure 4 shows at most 20% vertex reduction).

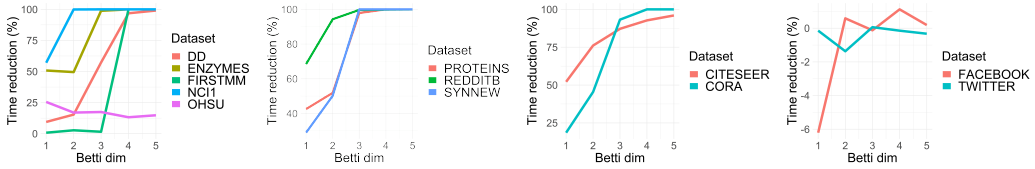

Figure 8: CoralTDA time reduction in graph and node classification datasets (higher is better). We do not observe a reduction in Facebook and Twitter datasets. The added computational cost of the algorithm results in negative gains.

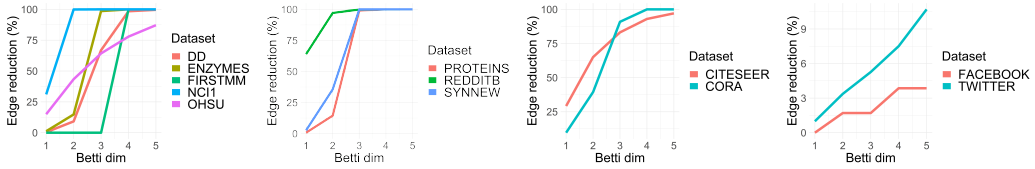

Figure 9: CoralTDA edge reduction $(100 \times (|\mathcal{E}| - |\mathcal{E}^k|)/|\mathcal{E}|)$ in graph and node classification datasets (higher is better). Reduction values are averages from graph instances of the datasets.

D.2 Clustering Coefficient and Higher Persistence Diagrams

While the proposed reduction algorithms provide a computationally efficient framework for higher-order topological summaries of graphs, in many real-world applications, we still need to address the important question of whether a given network exhibits a nontrivial higher $(k \geq 2)$ persistence diagram $PD_k(\mathcal{G})$. When the $(k+1)$-core $\mathcal{G}^{k+1}$ is sufficiently small, the answer immediately follows from our result. However, the problem is substantially more challenging in applications involving large complex networks such as blockchain transactions and protein interaction graphs. The goal is to assess whether for $k = 2, 3$, $PD_k(\mathcal{G})$ is trivial or not. Indeed, such higher homological features might be associated, for example, with money laundering patterns or drug-target interactions.

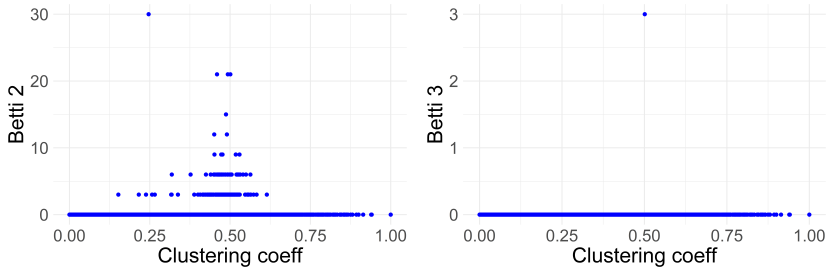

Figure 10: Clustering coefficients in kernel datasets. Betti 3 and higher do not exist in these graph datasets. In fact, even Betti 3 exists in a single graph only. Most Betti computations on such graphs can be avoided by using our conjecture on the clustering coefficient.

Our experiments show that many real-life data sets tend to exhibit nontrivial second and third persistence diagrams (see Figures 2 and 10). We compare these datasets by considering their clustering coefficients $CC(\mathcal{G})$. Our observations indicate that when $CC(\mathcal{G})$ is too low or too high, then the higher-order persistence diagrams associated with these data tend to be trivial. Intuitively, when $CC(\mathcal{G})$ is too low, the graph $\mathcal{G}$ tends to be too sparse to form a k-cycle and, hence, is unlikely to produce a nontrivial $PD_k(\mathcal{G})$. When $CC(\mathcal{G})$ is too high, then the graph $\mathcal{G}$ is too dense and every k-cycle is filled immediately by a $(k+1)$-complex in $\widehat{\mathcal{G}}$. We formulate this phenomenon in the form of the following conjecture.

Conjecture: For $k \geq 2$, there are $0 < \alpha_k < \beta_k < 1$ such that when $CC(\mathcal{G}) > \beta_k$ and $CC(\mathcal{G}) < \alpha_k$, then $PD_k(\mathcal{G}) = \emptyset$ with high probability.

However, clustering coefficient has the computational complexity of $O(|\mathcal{E}|^{1.48})$ [10], hence the using the coefficient may not seem as an improvement over coral reduction, which has a complexity of

$O(|\mathcal{E}| + |\mathcal{V}|)$. However, note that the clustering coefficient can be iteratively computed as the average of vertex clustering coefficients (in a parallel setting). In this approach, a stopping condition can be applied to terminate early when the coefficient can be approximated. The gain in computational time may help the coral technique on extremely large graphs or their induced subgraphs.

E Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See abstract for details.
   (b) Did you describe the limitations of your work? [Yes] See Section section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix D.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please see the proofs given in Appendix C.
   (b) Did you include complete proofs of all theoretical results? [Yes] Please see the proofs given in Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 6.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our experiments report averages with deviations (See Figure 6).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The experimental setting is given in Section 6.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use graph data only. Our datasets do not contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

F Broader Impact

The proposed methodology makes an important step toward addressing the major existing roadblock in bringing the powerful machinery of TDA to the analysis of large-scale networks, from online social

media to gene-to-gene interactions in bioinformatics. Undoubtedly, the application of TDA in learning such large-scale networks will have a substantial positive impact in a broad range of applications such as classification of anomalous subgraphs in blockchain transaction networks, discovering links between side effects and drugs, and drug re-purposing. However, the critical negative impact of the proposed methodology is associated with our current inability to accurately quantify the loss of topological information due to pruning and the influence of such loss on the final learning task. This is a fundamental question that needs to be addressed in the future.