Hypercore Decomposition for Non-Fragile Hyperedges: Full Supplementary Materials

[If a preview does not appear properly, please download the file]

I. Proofs

In this section, we provide the proofs with related analyses that are not presented in the main paper due to space limit.

A. Proof of Lemma 1: Differences between the $(k;l)$-hypercore and the (k,t)-hypercore

Definition 1 ($(k;l)$-hypercore (Def. 1 in [1]))**.** Given a hypergraph H and $k, l \in \mathbb{N}$, the $(k;l)$-hypercore of H denoted by $\tilde{C}_{k;l}(H)$, is the maximal subhypergraph of H such that each node in $\tilde{C}_{k;l}(H)$ has degree at least k and each hyperedge in $\tilde{C}_{k;l}(H)$ contains at least l nodes.

Lemma 1. Let $\tilde{C}_{k;l}(H)$ denote the $(k;l)$-hypercore of H. There exist H, k, t such that $C_{k,t}(H) \neq \tilde{C}_{k;l}(H)$ for any l.

The $(k; l = 2)$-hypercore is included in the proposed (k,t)-hypercore with $t = 0$ as an extremal case. There are some intuitions to see the differences between the $(k;l)$-hypercore and our proposed (k,t)-hypercore.

- When we obtain the $(k;l)$-hypercore of a given H with $l > 2$, all hyperedges of cardinality 2 are removed in the first place. Therefore, if we want to find an l such that $\tilde{C}_{k;l} = C_{k,t}$ where $C_{k,t}$ contains any hyperedge of cardinality 2, then we immediately have $l = 2$. Since $\tilde{C}_{k;2} = C_{k;0}$, we cannot find an l such that $\tilde{C}_{k;l} = C_{k,t}$ when $C_{k;0} \neq C_{k;t}$.
- Since the threshold in the (k,t)-hypercore is proportional, it imposes different absolute cardinality thresholds for hyperedges of different sizes. On the contrary, the $(k;l)$-hypercore imposes the same absolute cardinality threshold for each hyperedge.

We shall show two counterexamples from two intuitions above.

Proof. Consider $H = (V, E)$ with

$$E = \{\{1,2\}, \{1,3\}\{1,2,3,4\}, \{1,2,3,4,5\}\}.$$

The $(k = 2, t = 3/4)$-hypercore of H consists of the hyperedges

$$\{\{1,2\}, \{1,3\}, \{1,2,3\}\},$$

where the hyperedge $\{1,2,3,4,5\}$ is totally removed since only $3/5 < t = 3/4$ of the constituent nodes remain by the node-degree threshold $k = 2$. For the $(k;l)$-hypercore, when $l = 2$, the $(k = 2; l = 2)$-hypercore of H consists of the hyperedges

$$\{\{1,2\}, \{1,3\}, \{1,2,3,4\}, \{1,2,3,4\}\};$$

when $l = 3$ or $l = 4$, the $(k = 2; l)$-hypercore of H consists of the hyperedges $\{\{1,2,3,4\}, \{1,2,3,4\}\}$; when $l \geq 5$, the $(k = 2; l)$-hypercore of H is empty, completing the proof.

We show another counterexample. Consider $H = (V, E)$ with

$$E = \{\{1,2,3,4\}, \{1,2,5,6\}, \{5,6,7,8\}, \{3,4,9,10,11\}, \{1,2,3,4,5,6,7,8\}\}.$$

The $(k = 3, t = 1/2)$-hypercore of H consists of the hyperedges

$$\{\{1,2\}, \{1,2,5,6\}, \{5,6\}, \{1,2,5,6\}\}.$$

For the $(k;l)$-hypercore, when $l = 2$, the $(k = 3; l = 2)$-hypercore of H consists of the hyperedges

$$\{\{1,2,3,4\}, \{1,2,5,6\}, \{5,6\}, \{3,4\}, \{1,2,3,4,5,6\}\};$$

when $l \geq 3$, the $(k = 3; l)$-hypercore of H is empty, completing the proof. □

Remark 1. Our proposed (k,t)-hypercore allows arbitrarily fine-grained adjustment since the value of t is continuous in $[0, 1]$, while l must be an integer. In real-world hypergraphs, many hyperedges are of cardinality 2. Therefore, the $(k;l)$-hypercore with $l > 2$ is significantly less meaningful than the (k,t)-hypercore since many hyperedges are not taken into consideration at all. See Table I for the detailed number of hyperedges of different cardinality in each dataset we have used. See also Figures 8 to 21 for the performance of hypercoreness w.r.t $(k;l)$-hypercore to indicate the influence of nodes. Note again that the $(k; l = 2)$-hypercore is included in our proposed concept as the $(k, t = 0)$-hypercore. We observe that in most datasets, the $(k;l)$-hypercores become less meaningful and fail to indicate the influence of nodes when l becomes large.

TABLE I: **The number of hyperedges of different cardinality in each dataset.** For each dataset, we list the number of hyperedges of each specific size. Specifically, in most datasets, a large number of hyperedges are of cardinality 2. We use E_s to denote the set of hyperedges of cardinality s, for each s.

Dataset	$\lvert E\rvert$	$\lvert E_2\rvert$	$\lvert E_3\rvert$	$\lvert E_4\rvert$	$\lvert E_5\rvert$	$\lvert\bigcup_{s>5} E_s\rvert$
coauth-DBLP	2,169,663	693,364 (31.96%)	667,302 (30.76%)	419,431 (19.33%)	205,965 (9.49%)	183,601 (8.46%)
coauth-Geology	908,516	275,736 (30.35%)	227,950 (25.09%)	159,509 (17.56%)	99,140 (10.91%)	146,181 (16,09%)
NDC-classes	1,047	297 (28.37%)	121 (11.56%)	125 (11.94%)	94 (8.98%)	410 (39.16%)
NDC-substances	6,264	1,130 (18.04%)	745 (11.89%)	535 (8.54%)	500 (7.98%)	3,354 (53.54%)
contact-high	7,818	5,498 (70.32%)	2,091 (26.75%)	222 (2.84%)	7 (0.09%)	0 (0.00%)
contact-primary	12,704	7,748 (60.99%)	4,600 (36.21%)	347 (2.73%)	7 (0.09%)	0 (0.00%)
email-Enron	1,457	809 (55.53%)	317 (21.76%)	138 (9.47%)	63 (4.32%)	130 (8.92%)
email-Eu	24,399	12,753 (52.27%)	4,938 (20.24%)	2,294 (9.40%)	1,359 (5.57%)	3,055 (12.52%)
tags-ubuntu	145,053	28,138 (19.40%)	52,282 (36.04%)	39,158 (27.00%)	25,475 (17.56%)	0 (0.00%)
tags-math	169,259	25,253 (14.92%)	63,870 (37.74%)	50,892 (30.07%)	29,244 (17.28%)	0 (0.00%)
tags-SO	5,517,054	399,051 (7.23%)	1,537,702 (27.87%)	1,947,542 (35.30%)	1,632,759 (29.59%)	0 (0.00%)
threads-ubuntu	115,987	88,301 (76.13%)	21,621 (18.64%)	4,560 (3.93%)	1,117 (0.96%)	388 (0.33%)
threads-math	535,323	319,601 (59.70%)	142,065 (26.54%)	49,198 (9.19%)	16,402 (3.06%)	8,057 (1.51%)
threads-SO	8,589,420	5,210,916 (60.67%)	2,102,208 (24.47%)	787,701 (9.17%)	299,172 (3.48%)	189,423 (2.21%)

B. Proof of Proposition 1: Existence and uniqueness of the (k,t)-hypercores

Proposition 1. Given any hypergraph H, $k \in \mathbb{N}$, and $t \in [0,1]$, $C_{k,t}$ exists and is unique.

Proof. Since H is finite, the number of the subhypergraphs of H is also finite. Therefore, there must exist *one* subhypergraph with maximal total size (which is possibly an empty hypergraph) where each node has degree at least k and at least t proportion of the constituent nodes remain in each hyperedge, completing the proof of existence. To show the uniqueness, suppose the opposite, and let $C^1 = (V^1, E^1)$ and $C^2 = (V^2, E^2)$ be two distinct (k,t)-hypercores of H. Then we consider the hypergraph $C' = (V', E')$ with $E' = \{e_i^1 \cup e_i^2 : i \in I_{E^1} \cup I_{E^2}\}$. Clearly, C' is a subhypergraph of H with larger total size that satisfies the node-degree and hyperedge-fraction conditions, which contradicts the maximality and completes the proof. □

C. Proof of Proposition 2: Two-way containment of the (k,t)-hypercores

Proposition 2. Let H be any hypergraph. Fix any $k \in \mathbb{N}$, for any $0 \le t_1 < t_2 \le 1$, $C_{k,t_2}(H)$ is a subhypergraph of $C_{k,t_1}(H)$. Similarly, fix any $t \in [0,1]$, for any $k_1 < k_2 \in \mathbb{N}$, $C_{k_2,t}(H)$ is a subhypergraph of $C_{k_1,t}(H)$.

Proof. Suppose that $C_{k,t_2}(H)$ is not a subhypergraph of $C_{k,t_1}(H)$. Then we take the union $C_{k,t_2}(H) \cup C_{k,t_1}(H)$ and we obtain a hypergraph that is strictly larger than $C_{k,t_1}(H)$ and satisfies the conditions of (k,t_1)-hypercore, which contradicts with the maximality, completing the proof. The second statement can be proved in a similar way. □

D. Proof of Theorem 1: Correctness and time complexity of Algorithm 1

Theorem 1. Given $H = (V, E)$, $k \in \mathbb{N}$, and $t \in [0,1]$, Algorithm 1 returns $C_{k,t}(H)$ in $O(|V| + |E| + (1-t)\sum_{e\in E}|e|)$ time.

Proof. **Correctness.** The size of a hyperedges changes only when some node in $\mathcal{R}$ is removed from it, and the degree of a node changes only when some incident hyperedge is removed. Therefore, when Algorithm 1 ends, each node has degree at least k, otherwise it must have been included in $\mathcal{R}$ and removed, and each hyperedge satisfies the hyperedge-fraction condition, otherwise it must have been removed. This implies that the output of Algorithm 1 satisfies both the node-degree and hyperedge-fraction conditions w.r.t H, k, and t. We now show the maximality. Suppose not, and let (v, e_i) be the first node-hyperedge pair that appears during the process of Algorithm 1 with $v \in e_i \in E(C_{k,t})$ but $v \notin e_i' \in E'$, where $C' = (V', E')$ is the returned hypergraph. This implies that v is removed from e, and thus v is included in $\mathcal{R}$ because its degree has been below k. However, by the definition of the (k,t)-hypercore and the assumption that (v, e_i) is the first pair, before the deletion, the degree of v is at least k, which completes the proof by contradiction.

Time complexity. We assume the input hypergraph has been loaded in the memory and thus do not count the complexity of loading the hypergraph. Checking the initial degrees (Line 1) takes $O(|V|)$. In the while loop, each node is added into the set of nodes to be removed at most once since each node is added exactly when its degree decreases from k to $k-1$. Therefore, this process takes $O(|V|)$. By checking the incident edges of each node in $\mathcal{R}$, we find all e_i's intersecting with $\mathcal{R}$, which takes $O(|\mathcal{R}|) = O(V)$ in total. Hash tables are used to implement the sets. Before a hyperedge $e \in E$ is totally removed, it can be visited at most $|e| - \max(\lceil t|e|\rceil, 2) + 1$ times. This process takes $O(\sum_{e\in E}(|e| - \max(\lceil t|e|\rceil, 2) + 1)) = O(|E| + (1-t)\sum_{e\in E}|e|)$, completing the proof. □

E. Proof of Theorem 2: Correctness and time complexity of Algorithm 2

Theorem 2. Given $H = (V, E)$ and $t \in [0, 1]$, Algorithm 2 returns $c_t(v)$ for all $v \in V$ in $O(c_t^*|V| + |E| + (1-t)\sum_{e\in E}|e|)$ time.

Proof. **Correctness.** For each node v, the assignment of $c_t(v)$ happens only once when $v \in \mathcal{R}$, i.e., before its deletion. By Theorem 1, $c_t(v) = k-1$ implies that v is not in the (k,t)-hypercore but in the previous (k',t)-hypercore where each node has degree at least $k-1$, i.e., v is in the $(k-1,t)$-hypercore, completing the proof.
Time complexity. The values of k increases $O(c_t^*)$ times, thus the process in Lines 4 and 5 is repeated for $O(c_t^*)$ times and takes $O(c_t^*|V|)$. The assignment of t-hypercoreness of each node (Line 7) takes $O(|V|)$. As shown in the proof of Theorem 1, each hyperedge is visited at most $|e| - \max(\lceil t|e| \rceil, 2) + 1$ times before being deleted and each node is added to the set of nodes to be removed only once. Therefore, the remaining process takes $O(|V| + |E| + (1-t)\sum_{e\in E}|e|)$, completing the proof. □

F. Proof of Theorem 3: Correctness and time complexity of Algorithm 3

Theorem 3. Given $H = (V, E)$ and $k \in \mathbb{N}$, Algorithm 3 returns $f_k(v)$ for all $v \in V$ in $O(\sum_{e\in E}|e|)$ time.

Proof. **Correctness.** For each node v, the assignment of $f_k(v)$ happens only once when $v \in \mathcal{R}$, i.e., before its deletion. By Theorem 1, $f_k(v) = t$ implies that v is in the (k,t)-hypercore with degree k and is in at least one hyperedge that is in the (k,t)-hypercore but not in any (k,t')-hypercore with $t' > t$. Therefore, v is not in any (k,t')-hypercore with $t' > t$, completing the proof.
Time complexity. Recording the hyperedge sizes (Line 1) takes $O(|E|)$. By Theorem 1, computing $C_{k,0}$ (Line 2) takes $O(\sum_{e\in E}|e|)$. As shown in the previous proofs, the while loop (Lines 4 to 12) takes $O(|V| + |E| + \sum_{e\in E}|e|) = O(\sum_{e\in E}|e|)$, completing the proof. □

II. Extended related work

In this section, we introduce some extended related publications on similar topics, e.g., the generalizations and applications of the k-cores.

- Zhang et al. [2] (ICDE'12) generalize the k-cores to the triangle k-cores, which are essentially the k-trusses, to extract the information in pairwise graph.
- Peng et al. [3] (ICDE'18) generalize the k-cores to uncertain graphs, where each edge exists in a probabilistic way. Specifically, they consider the problem of k-core decomposition on uncertain graphs, and propose the (k,θ)-core, which consists of the nodes with probability at least θ to be in the k-core of the given uncertain graph.
- Wang et al. [4] (ICDE'18) generalize the k-cores to geo-social networks. Specifically, they propose the radius-bounded k-core by taking the spatial constraints into consideration, where they require every two nodes in the radius-bounded k-core to have a Euclidean space less than some given threshold.
- Zhang et al. [5] (ICDE'20) generalize the k-cores to the (k,p)-cores. Specifically, they further require each node in the (k,p)-core to have at least p fraction of its neighbors in the (k,p)-core too.
- Bonchi et al. [6] (SIGMOD'19) generalize the k-cores to the (k,h)-cores. Specifically, they relax the node-degree condition by requiring each node in the (k,h)-cores to have at least k other nodes at distance at most h, i.e., to have at least k h-hop neighbors. Dai et al. [7] (CIKM'21) further investigate the (k,h)-cores.
- Zhang et al. [8] (PVLDB'17) generalize the k-cores to the (k,r)-cores, where they take the similarity between each pair of nodes w.r.t the attributes also into consideration. They use the proposed model to find cohesive subgraphs in real-world graphs.
- Liu et al. [9] (VLDBJ'20) and Ding et al. [10] (CIKM'17) consider the computation of the (α,β)-cores in bipartite graphs. Similar to k-cores, in the (α,β)-core, each node in the first partition is required to have at least α neighbors in the second partition, while each node in the second partition is required to have at least β neighbors in the first partition.
- Shin et al. [11] (CIKM'16) apply the k-cores to real-world graphs, in order to find the structural patterns, detect anomalies. They also provide algorithms based on the observations to solve existing problems with high efficiency and comparable performance.
- Victor et al. [12] (KDD'21) generalize the k-cores by combining multiple node properties and introducing the notion of data depth. The proposed method is able to simultaneously handle multiple node and edge attributes.
- Chen et al. [13] (TKDE'21) generalize the k-trusses to the (k,τ)-trusses, by taking the h-hop neighbors of each node into consideration, which is based on a similar idea of the (k,h)-cores in [6].
- Sarıyüce et al. [14] (WSDM'18) propose to find the dense substructures in bipartite graphs by a peeling algorithm, which is similar to the standard algorithm to obtain the k-core. Instead of focusing on pairwise connectivity, they define the dense model based on the butterfly motifs, i.e., $(2,2)$-bicliques.

TABLE II: **The running time (in seconds) of (k,t)-hypercore computation.** We run the (k,t)-hypercore computation with different values of k and t, on the relatively large datasets we have used. Average running time over five independent trials is reported.

(k,t)	coauth-DBLP	coauth-Geology	tag-SO	threads-SO
$(5,0)$	0.128	0.062	0.029	0.327
$(5,0.2)$	0.129	0.063	0.029	0.327
$(5,0.4)$	0.135	0.067	0.029	0.327
$(5,0.6)$	0.165	0.082	0.029	0.330
$(5,0.8)$	0.275	0.125	0.029	0.452
$(5,1)$	0.305	0.139	0.032	0.481
$(10,0)$	0.184	0.079	0.029	0.561
$(10,0.2)$	0.185	0.079	0.030	0.562
$(10,0.4)$	0.192	0.085	0.030	0.563
$(10,0.6)$	0.244	0.111	0.029	0.580
$(10,0.8)$	0.346	0.139	0.037	0.853
$(10,1)$	0.333	0.130	0.040	0.909
$(20,0)$	0.242	0.104	0.034	0.799
$(20,0.2)$	0.249	0.097	0.034	0.800
$(20,0.4)$	0.257	0.107	0.034	0.802
$(20,0.6)$	0.311	0.132	0.034	0.861
$(20,0.8)$	0.307	0.128	0.037	1.264
$(20,1)$	0.290	0.118	0.040	1.330

Fig. 1: **HyCoM+ has linear scalability w.r.t the budget and the hypercore size.** On the left, we show the running time of HyCoM-1 and HyCoM+ with b increasing. On the right, we show the running time of HyCoM-1 and HyCoM+ while upscaling the tags-SO dataset ($b = 100$). HyCoM+ takes less than 10 minutes (574 seconds) when the total size of the input hypergraph is 1.37B ($64\times$ upscaled).

- Gabert et al. [15] (WSDM'21) make use of a generalization of the k-cores and the k-trusses, the k-nuclei, to detect dense substructures. Specifically, for each s-clique in the graph, they consider the number of t-cliques containing it, where $s < t$ are two parameters. They also establish the connection between the k-nuclei to the k-cores in hypergraphs.
- Preti et al. [16] (WWW'21) generalize the k-trusses to simplicial complexes, a special kind of hypergraphs, and use the proposed model to study the structure of real-world simplicial complexes.

III. Running time analysis

In this section, we show the practical running time of the computation of the (k,t)-hypercores with different parameters, i.e., different values of k and t, on the real-world hypergraphs. We also provide more running time results to show the scalability of (k,t)-hypercore computation and HyCoM.

A. (k,t)-hypercore computation with different parameters

In Table II, we show the running time of (k,t)-hypercore computation with different values of k and t.

B. Scalability

In Fig. 1, we show the linear scalability w.r.t the budget and the hypercore size of HyCoM and HyCoM+, where we generate synthetic hypergraphs by upscaling the original ones. In particular, we duplicate each hyperedge up to $64\times$, which is simple and generates realistic hypergraphs.

In Table III, we show the running time of (k,t)-hypercore computation and HyCoM-1 / HyCoM+ on the synthetic hypergraphs generated by scaling up the tags-SO dataset up to $64\times$ using the HyperCL algorithm mentioned in [17].

In Table IV, we show the running time of (k,t)-hypercore computation on the synthetic hypergraphs with 1000 nodes and increasing densities. Specifically, we generate hypergraphs with 50% of the hyperedges of size 2, 20% of size 3, 20% of size 4, 10% of size 5. For the hyperedges of each size, we choose them uniformly at random among all possible subsets. We start from 1000 nodes and 25000 hyperedges, which is in a similar scale with the real-world dataset *email-Eu*. We upscale the

TABLE III: **The running time (in seconds) of (k,t)-hypercore computation, HyCoM-1 and HyCoM+ on the synthetic large-scale hypergraphs.** We scale up the real-world dataset, tags-SO, up to $64\times$, using the HyperCL algorithm mentioned in [17]. The parameters $k = 10, t = 0.6$, and budget $b = 100$ are consistently used.

Scale factor	$\|V\|$	$\|E\|$	$\sum_{e\in E}\|e\|$	hypercore computation	HyCoM+	HyCoM
$1\times$	50K	5.5M	21M	0.029	31.732	148.364
$2\times$	100K	11M	42M	0.059	64.612	310.157
$4\times$	200K	22M	84M	0.122	117.342	597.149
$8\times$	400K	44M	168M	0.249	196.911	976.745
$16\times$	800K	88M	336M	0.499	287.560	1421.297
$32\times$	1.6M	176M	672M	1.034	392.870	2048.148
$64\times$	3.2M	352M	1.3B	2.004	519.081	3063.616

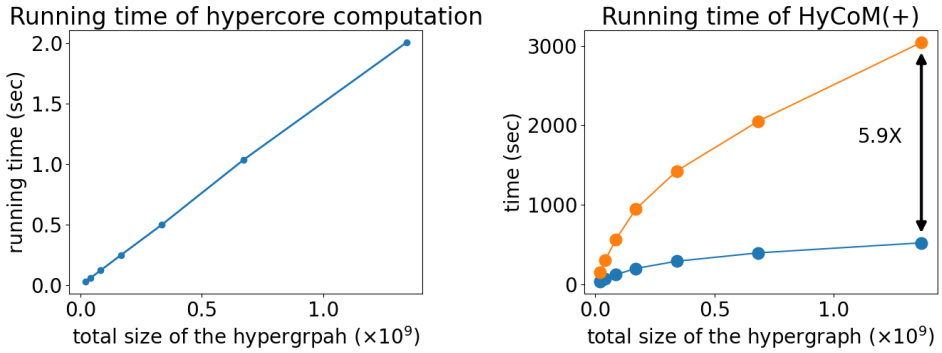
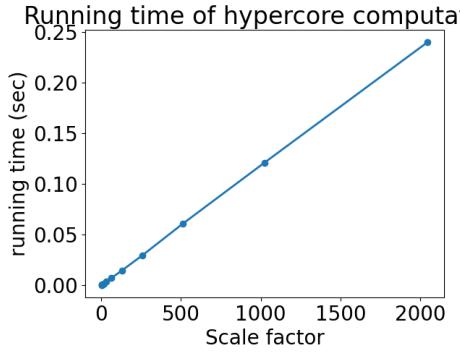


TABLE IV: **The running time (in seconds) of (k,t)-hypercore computation on synthetic hypergraphs with different densities.** All synthetic hypergraphs contain 1000 nodes. The parameters $k = 10, t = 0.6$ are consistently used.

Scale factor	$\|E\|$	$\sum_{e\in E}\|e\|$	hypercore computation
$1\times$	25K	72.5K	0.000115
$2\times$	50K	145K	0.000226
$4\times$	100K	290K	0.000445
$8\times$	200K	580K	0.000885
$16\times$	400K	1.16M	0.00178
$32\times$	800K	2.32M	0.00356
$64\times$	1.6M	4.64M	0.00716
$128\times$	3.2M	9.28M	0.0144
$256\times$	6.4M	18.56M	0.0293
$512\times$	12.8M	37.12M	0.0608
$1024\times$	25.6M	74.24M	0.121
$2048\times$	51.2M	148.48M	0.240



number of hyperedges up to $2048\times$. Notably, the simple duplication we have used can also be seen as a way to increase the density without changing the number of nodes.

IV. Details of Datasets

In this section, we provide more details of the datasets that we have used in our experiments.

- **coauth-DBLP/Geology.** In these two *coauthorship* hypergraphs, each hyperedge represents a publication, and the constituent nodes of a hyperedge represent the authors of the corresponding publication.
- **NDC-classes/substances.** In these two hypergraphs from the *National Drug Code (NDC) Directory*, each hyperedge represents a drug (with its unique NDC code), and the constituent nodes of a hyperedge represent the class labels (for NDC-classes) or the ingredients (for NDC-substances) of the drug.
- **contact-high/primary.** In these two *contact* hypergraphs, each hyperedge represents a group of people with pairwise interaction within a specific time interval, where each node is a person.
- **email-Enron/Eu.** In these two *email* hypergraphs, each hyperedge represents an email (possibly sent to multiple people individually at the same time), which contains the sender and all the receivers as its constituent nodes.
- **tags-ubuntu/math/SO.** In these three *tags* hypergraphs from https://stackoverflow.com/, each node represents a tag, and each hyperedge represents a question, where each constituent node represents a tag applied to the question.
- **threads-ubuntu/math/SO.** In these three *threads* hypergraphs also from https://stackoverflow.com/, each hyperedge represents a thread, where each constituent node represents a person that participates in it.

We have used the preprocessed version of the datasets where each hyperdege consists of at most 25 nodes. Notably, on https://www.cs.cornell.edu/~arb/data/, the *full* version of the datasets, in which the cardinality of the hyperedges is not limited, is also available.

V. Hypercore sizes and densities

In this section, we give the detailed definition of the hypercore-size-mean-difference (HSMD) distance, and provide the full results regarding the (k,t)-hypercore sizes. We also discuss the densities of hypercores.

A. Definition of HSMD distance

Given any hypergraph $H=(V,E)$, by the containment properties (Proposition 2 in the main paper), $1 \le c_t^* \le c_0^*, \forall t$. Therefore, we can use the normalizer $\mathcal{N}_H : [0,1] \to \{1,2,\dots,c_0^*\}$ defined by $\mathcal{N}_H(x) = \lceil (c_0^*)^x \rceil$. We then define the dissimilarity between two hypercore sizes by their difference in log scale (as in Figure 2 in the main paper), which is also normalized in $[0,1]$. Formally, the dissimilarity between two hypergraphs H_1, H_2 at the normalized point (x,t) with $x,t \in [0,1]$ is $\tilde{d}(x,t;H_1,H_2) \coloneqq \min(|\tilde{n}_{\mathcal{N}_{H_1}(x),t}(H_1) - \tilde{n}_{\mathcal{N}_{H_2}(x),t}(H_2)|, 1)$, where we let $\tilde{n}_{k,t} = -1$ if $C_{k,t}$ is empty. This dissimliarity can also be understood as the difference between the same position of two subfigures in Figure 2 in the main paper. Finally, we define the hypercore-size-mean-difference (HSMD) distance, which lies between 0 and 1, as follows:

Definition 2 (HSMD distance)**.** Given two hypergraphs H_1 and H_2, the hypercore-size-mean-difference (HSMD) distance between H_1 and H_2 is defined as

$$\mathrm{HSMD}(H_1,H_2) \coloneqq \sqrt{\int_0^1 \int_0^1 (\tilde{d}(x,t;H_1,H_2))^2 \, dx \, dt}.$$

B. Full results

In Figure 2, we provide the full results regarding the (k,t)-hypercore sizes.

C. Hypercore densities

In Fig. 3, for each dataset and each $t \in [0,1]$, we show the relative density of the (c_t^*,t)-hypercore, which is defined as $\tilde{\delta}_t = \delta(C_{c_t^*,t})/\delta(H)$. Note that the hypercores are significantly denser than the whole hypergraph, especially when t is small. In addition, except for the tags-SO dataset, similarity between hypergraphs in the same domain is observed.

Similarly to the HSMD distance (Definition 2), we define the relative-density-mean-difference (RDMD) distance between two hypergraphs.

Definition 3 (RDMD distance)**.** Given two hypergraphs H_1 and H_2, the relative-density-mean-difference (RDMD) between H_1 and H_2 is defined as

$$\mathrm{RDMD}(H_1,H_2) \coloneqq \sqrt{\int_0^1 (\log \tilde{\delta}_t(H_1) - \log \tilde{\delta}_t(H_2))^2 \, dt}.$$

In Figure 4, we report the RDMD distance between each pair of datasets (except for the tags-SO dataset), where the small RDMD distance between hypergraphs in the same domain is clearly observed.

We show an example when the density of (k,t)-hypercore with maximum k does not decrease when t increase. Consider the hypergraph with $E = \{\{1,2,6\},\{1,2,7\},\{3,4,5,8,9,10\},\{3,4,5,11,12,13\}\}$. When $t=1/2$, the (k,t)-hypercore with maximum $k=2$ contains hyperedges $\{\{1,2\},\{1,2\},\{3,4,5\},\{3,4,5\}\}$, and the density is $4/5 = 0.8$; when $t=2/3$, the (k,t)-hypercore with maximum $k=2$ contains hyperedges $\{\{1,2\},\{1,2\}\}$, and the density is $2/2=1$.

TABLE V: **The detailed statistics on the heavy-tailed distribution tests.** For each dataset and each $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, we report the log-likelihood ratio (R-value) of heavy-tailed distributions against the exponential distribution with its p-value. In most cases, the log-likelihood ratio is positive and the p-value is small, which implies the significance of the heavy-tailed distributions. Left: R-value; Right: p-value.

Dataset	$t = 0$	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1$
coauth-DBLP	156.75 / 7.24e-13	184.96 / 7.28e-16	139.27 / 3.36e-13	50.80 / 0.001	1685.14 / 1.45e-40	117.76 / 4.75e-56
coauth-Geology	106.38 / 3.06e-11	83.21 / 7.70e-8	31.52 / 6.86e-8	17.80 / 9.57e-5	1049.01 / 0.0	989.44 / 0.0
NDC-classes	45.32 / 4.38e-6	364.85 / 5.80e-24	103.93 / 4.90e-14	282.71 / 1.06e-42	290.16 / 4.21e-45	242.25 / 1.28e-41
NDC-substances	30.90 / 2.34e-5	26.95 / 0.00061	2608.99 / 4.33e-208	1884.07 / 6.54e-171	1175.06 / 8.51e-91	221.15 / 2.78e-24
contact-high	16.15 / 3.20e-20	-	-	16.76 / 0.0040	0.70 / 0.48	0.70 / 0.48
contact-primary	0.19 / 0.23	-	-	136.51 / 1.75e-16	127.81 / 6.41e-13	127.81 / 6.41e-13
email-Enron	0.29 / 0.73	2.05 / 0.24	2.67 / 0.063	8.43 / 0.024	1.55 / 2.4e-267	0.22 / 0.76
email-Eu	-0.47 / 0.60	0.05 / 0.97	2.40 / 3.77e-9	83.69 / 2.05e-11	-0.28 / 0.36	11.26 / 0.005
tags-ubuntu	201.24 / 4.28e-21	-	-	83.69 / 2.05e-11	-14.68 / 5.76e-6	-17.40 / 1.30e-32
tags-math	8.81 / 0.06	-	-	15.31 / 0.027	-17.96 / 1.03e-9	-14.34 / 0.00052
tags-SO	616.59 / 2.41e-29	-	-	3617.24 / 8.07e-222	2189.25 / 5.27e-234	-17.40 / 1.30e-32
threads-ubuntu	279.41 / 2.09e-22	278.50 / 2.71e-22	259.53 / 8.96e-22	130.95 / 1.15e-14	119.14 / 8.15e-14	226.74 / 6.06e-41
threads-math	226.30 / 1.45e-23	225.66 / 1.68e-23	5192.50 / 2.08e-282	11461.10 / 0.0	3305.64 / 0.0	6632.53 / 0.0
threads-SO	444.93 / 3.47e-57	436.37 / 4.84e-56	153.14 / 6.50e-19	-23.83 / 7.24e-8	6002.46 / 0.0	2682.90 / 4.64e-102

VI. Correlations

In Figure 5, we provide the full results regarding the Pearson's r between t-hypercoreness sequences with different t values and each of the **degree** and **coreness** sequences in the *unweighted* and *weighted* clique expansions. We also report Pearson's r between each pair of t-hypercoreness sequences.

VII. Heavy-tailed distributions

In Table V, we provide the full results on the heavy-tailed distribution tests. Specifically, we report the log-likelihood ratio (R-value) of heavy-tailed distributions against the exponential distribution, where a positive R-value indicates that heavy-tailed distributions are more promising; and the p-values, where a small p-value indicates that the heavy-tailed or exponential distribution is significant.

In Figure 6, for the NDC-classes and threads-ubuntu datasets and $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, we show the numbers of nodes with t-hypercoreness at least k with different k values, together with the results of power-law fitting, i.e., linear regression in log-log scale; and consistent power-law distributions of the t-hypercoreness sequences are observed.

VIII. Information gain

In Figure 7, we provide the full results regarding the information gain.

IX. Influential-node identification

In Figures 8 to 21, we provide the full results regarding influential-node identification.

Definition 4 (clique expansion)**.** Given a hypergraph $H = (V, E)$, its unweighted clique expansion is $G_{\text{uc}}(H) = (V, \mathcal{E})$, and its weighted clique expansion is $G_{\text{wc}}(H) = (V, \mathcal{E}, \omega)$, where the edge set $\mathcal{E} = \{(u, v) \in \binom{V}{2} : \exists e \in E \ s.t. \ \{u, v\} \subseteq e\}$, and the weight function $\omega((u, v)) = |\{i \in I_E : \{u, v\} \subseteq e_i\}|$.

Clique expansion is a simple way to convert a hyperedge into a pairwise graph, so that the existing methods on pairwise graph analysis can be directly utilized. However, it is easy to see that the higher-order interactions in the hypergraph, i.e., the interactions among at least three nodes, are irreversibly lost due to the nature of pairwise graphs. This information loss is natural since for a fix set of nodes V with $|V| = n$, there are $\binom{n}{2} = O(n^2)$ possible pairs consisting of two nodes in V, while there are $2^n - n - 1 = O(2^n)$ possible subsets of cardinality at least 2 consisting of nodes in V. In many existing works [18]–[31], this information loss has been investigated and discussed, and many trials have been done to practically extract the higher-order interactions in hypergraphs.

Lemma 1. For two different hypergraphs $H_1 \neq H_2$, $G_{\text{wc}}(H_1) = G_{\text{wc}}(H_2)$ possibly holds[1].

Proof. A simplest example can be from the fact that the weighted clique expansion of a pairwise graph is itself. By this fact, any non-pairwise hypergraph has the same weighted clique expansion with its weighted clique expansion, completing the proof.

Other simple examples can also be easily constructed. Consider H_1 with $E(H_1) = \{\{2, 3\}, \{1, 2, 3, 4\}\}$ and H_2 with $E(H_2) = \{\{1, 4\}, \{1, 2, 3\}, \{2, 3, 4\}\}$. It is easy to see that $G_{\text{wc}}(H_1) = G_{\text{wc}}(H_2)$, completing the proof. □

[1]Note that this also implies that $G_{\text{uc}}(H_1) = G_{\text{uc}}(H_2)$.

TABLE VI: **The summary of the results on influential-node identification.** We report the R^2 values between each quantity and node-influence in each setting. Some quantities are emitted due to their consistently weak performance. In most cases, the t-hypercoreness with a specific t shows the best performance. Abbreviations: HC = hypercoreness, C = coreness, U = unweighted, W = weighted, EC = eigencentrality, D = degree.

Dataset	t-HC (best t)	(t=0/l=2)-HC	l-HC (best l>2)	C-U	C-W	Cp-U (best p)	Cp-W (best p)	EC-U	EC-W	D
coauth-DBLP	**0.87 (13/25)**	0.81	0.80 (3)	0.50	0.76	0.54 (0.75)	0.76 (0.75)	0.19	0.01	0.56
coauth-Geology	**0.87 (3/7)**	0.83	0.84 (3)	0.55	0.78	0.59 (0.75)	0.79 (0.75)	0.08	0.02	0.62
NDC-classes	0.88 (3/5)	0.53	0.89 (9)	0.75	**0.91**	0.74 (0.75)	0.90 (0.75)	0.78	0.42	0.53
NDC-substances	**0.93 (5/8)**	0.61	0.88 (8)	0.74	0.87	0.74 (0.75)	0.87 (0.25)	0.71	0.34	0.50
contact-high	**0.90 (2/3)**	**0.90**	0.72 (3)	0.86	0.84	0.82 (0.75)	0.80 (0.75)	0.58	0.25	0.62
contact-primary	**0.95 (2/3)**	**0.95**	0.75 (3)	0.77	0.93	0.73 (0.75)	0.93 (0.75)	0.56	0.47	0.62
email-Enron	**0.92 (8/15)**	0.85	0.77 (3)	0.53	0.82	0.53 (0.75)	0.79 (0.75)	0.45	0.41	0.67
email-Eu	**0.96 (2/3)**	0.89	0.88 (3)	0.73	0.79	0.73 (0.5)	0.80 (0.75)	0.67	0.21	0.69
tags-ubuntu	**0.94 (1)**	0.52	0.85 (4)	0.91	0.67	0.91 (0.75)	0.67 (0.75)	0.82	0.24	0.27
tags-math	**0.98 (1)**	0.60	0.94 (4)	0.85	0.76	0.85 (0.75)	0.76 (0.75)	0.87	0.24	0.33
tags-SO	**0.93 (2/3)**	0.87	0.88 (3)	0.91	0.88	0.91 (0.75)	0.88 (0.75)	0.49	0.16	0.27
threads-ubuntu	**0.88 (2/3)**	0.80	0.57 (3)	0.86	0.79	0.81 (0.75)	0.77 (0.75)	0.62	0.13	0.30
threads-math	**0.94 (2/3)**	0.77	0.91 (3)	**0.94**	0.85	**0.94 (0.75)**	0.85 (0.75)	0.73	0.23	0.26
threads-SO	**0.93 (2/3)**	0.85	0.88 (3)	0.91	0.89	0.91 (0.75)	0.88 (0.75)	0.55	0.32	0.45
Average	**0.92**	0.77	0.83	0.77	0.82	0.77	0.82	0.58	0.25	0.48
# Best*	**13**	2	0	1	1	1	0	0	0	0

* the number of datasets where each quantity is most indicative.

Definition 5 (p-clique expansion)**.** Given a hypergraph $H = (V, E)$ and $p \in (0, 1]$, the unweighted p-clique expansion of H is a (random) unweighted pairwise graph $G_{\mathrm{uc};p}(H) = (V, \mathcal{E})$ generated by keeping each edge in the unweighted clique expansion of H with probability p, independently at random; and the weighted p-clique expansion of H is a (random) weighted pairwise graph $G_{\mathrm{wc};p}(H) = (V, \mathcal{E}, \omega)$, where the (random) weight function $\omega_p((u, v)) \sim \mathcal{B}(|\{i \in I_E : \{u, v\} \subseteq e_i\}|, p)$ is generated by a Bernoulli process, i.e., follows a binomial distribution.

Remark 2. When $p = 1$, the p-clique expansions recover the standard clique expansions.

In Figures 8 to 21, we also show the performance of the *l-hypercoreness* w.r.t the (k, l)-hypercore (Definition 1) and the p-coreness w.r.t the unweighted (p-coreness-U) and weighted (p-coreness-W) p-clique expansions (Definition 5) in influential-node identification. Note again that the $(k; l = 2)$-hypercore is included in our proposed concept as the $(k, t = 0)$-hypercore. We observe that in most datasets, the $(k; l)$-hypercores become less meaningful and fail to indicate the influence of nodes when l becomes large. In Table VI, we summarize the results on influential-node identification. In most cases, the t-hypercoreness with a specific but nontrivial t shows the best performance.

X. Hypergraph vulnerability detection

In Table VII, we provide the full results regarding the collapsed (k, t)-hypercore problem.

XI. Results on more datasets

In this section, we show the results on six additional datasets other than the datasets obtained from cs.cornell.edu/~arb/data which we have used in the main paper. Unlike the datasets we have used in the main paper, those additional datasets do not have clearly-defined domains. Although this makes it difficult to validate some domain-based observations we have presented in the main paper, it is interesting to check which domains those additional datasets are similar to. The basic statistics of those additional datasets are reported in Table VIII. Like the datasets used in the main paper, we only keep hyperedges of sizes between 2 and 25 and do not allow repeated hyperedges. Specifically, the additional datasets are:

- **linux-kernel-mailing** [32]:[2] This hypergraph represents the contributions by users to threads on the Linux kernal mailing list. Each node is a user and each hyperedge is a thread. A node v is in a hyperedge e if the user v contributed to the thread e.
- **marvel** [33]:[3] This hypergraph represents the collaboration in the Marvel Universe. Each node is a character and each hyperedges is a Marvel comic book. A node v is in a hyperedge e if the character v appears in the comic book e.
- **foursquare-NYC-restaurant** [34]:[4] This hypergraph represents the digital footprints of restaurants in NYC collected from Foursquare from 24 October 2011 to 20 February 2012. Each node is a user and each hyperedges is a restaurant. A node v is in a hyperedge e if the user v visited the restaurant e.

[2]http://konect.cc/networks/lkml_person-thread

[3]http://bioinfo.uib.es/~joemiro/marvel/porgat.txt

[4]https://sites.google.com/site/yangdingqi/home/foursquare-dataset (including the other two *foursquare* datasets)

- **foursquare-NYC** [35]: This hypergraph represents the check-ins in New York city collected from Foursquare from 12 April 2012 to 16 February 2013. Each node is a user and each hyperedges is a venue (not necessarily restaurants). A node v is in a hyperedge e if the user v visited the venue e.
- **foursquare-Tokyo** [35]: This hypergraph represents the check-ins in Tokyo city collected from Foursquare from 12 April 2012 to 16 February 2013. Each node is a user and each hyperedges is a venue (not necessarily restaurants). A node v is in a hyperedge e if the user v visited the venue e.
- **wikinews** [36]:[5] This hypergraph represents the edits in Wikipedia pages about news events. Each node is a user and each hyperedges is a page. A node v is in a hyperedge e if the user v edited the page e.

A. Hypercore sizes and densities

In this subsection, we provide the results regarding the (k,t)-hypercore sizes for the additional datasets (see Figure 22). In Table IX, we report the full results w.r.t HSMD distances between each pair of datasets including the datasets used in the main paper as well as the six additional datasets. We want to check:

- Q1: Which domain(s) is each additional dataset similar to?
- Q2: How similar is each pair of the additional datasets (especially *foursquare-NYC* and *foursquare-Tokyo* which are clearly from the same domain)?

We first analyze Q1:

- the *linux-kernel-mailing* dataset has the smallest HSDM distances with the two *email* datasets (which is consistent with the domain its name "mailing" suggests), followed by the *tags* datasets
- the *marvel* dataset has relatively high HSDM distances with all the benchmark datasets used in the main text; the closest ones are the two *NDC* datasets
- the *foursquare-NYC-restaurant* dataset has relatively high HSDM distances with all the benchmark datasets used in the main text; the closest ones are the two *coauth* datasets
- the *foursquare-NYC* and *foursquare-Tokyo* datasets have the smallest HSDM distances with the two *email* datasets
- the *wikinews* dataset has relatively high HSDM distances with all the benchmark datasets used in the main text; the closest ones are the two *contact* datasets

We then analyze Q2:

- Overall, the additional datasets are quite different with each other w.r.t HSDM distances
- Interestingly, we can see that the *foursquare-NYC* and *foursquare-Tokyo* datasets which are clearly from the same domain share very low HSDM distance, which is consistent with our observation presented in the main text
- Besides, the *linux-kernel-mailing* dataset and the *foursquare-NYC* and *foursquare-Tokyo* datasets also share relatively low HSDM distances

B. Heavy-tailed distributions

This subsection corresponds to the "Distributions of t-hypercoreness" subsection in the main text. In Figure 23, for each t value, we report the log-likelihood ratio (R-value) of heavy-tailed distributions against the exponential distribution. In particular, we compute the log-likelihood ratio for two heavy-tailed distributions, power-law and log-normal, and take the maximum. In most cases, the log-likelihood ratio is positive, which supports the possibility that the t-hypercoreness follows heavy-tailed distributions consistently regardless of the value of t. Moreover, strong power-law distributions are observed in some datasets. In Figure 24, for each additional dataset and each t value, we show the R^2 value of the power-law fitting w.r.t the numbers of nodes with t-hypercoreness at least k with different k values.

C. Correlations and information gain

This subsection corresponds to the "Heterogeneity of t-hypercoreness" subsection in the main text.

Correlations. To show (a) the distinctiveness of t-hypercoreness from existing centrality measures, and (b) the dissimilarity between t-hypercoreness with different t values, we measure the Pearson correlation coefficients.

For the additional datasets, in Figure 25, we report Pearson's r between the t-hypercoreness sequences with different t values and each of the **degree** and **coreness** sequences in the *unweighted* and *weighted* clique expansions. We also report Pearson's r between each pair of t-hypercoreness sequences. As observed in the main text, even for the same hypergraph, the hypercoreness sequences with different t values can be fairly dissimilar, which still holds for the additional datasets.

Information gain. We also show from the perspective of information theory that hypercoreness sequences with different t values contain different information. For the additional datasets, in Figure 26, we report the information gain for different t values. As observed in the main text, the highest information gain is achieved by different t values in different datasets. Also, in Table X, we report the full results w.r.t Pearson's correlation coefficient between each pair of datasets including the datasets

[5]http://konect.cc/networks/edit-enwikinews

used in the main paper as well as the six additional datasets. Notably, we observe that the *foursquare-NYC* and *foursquare-Tokyo* datasets share very high Pearson's correlation coefficient, validating the domain-based similarity that we have observed and presented in the main text.

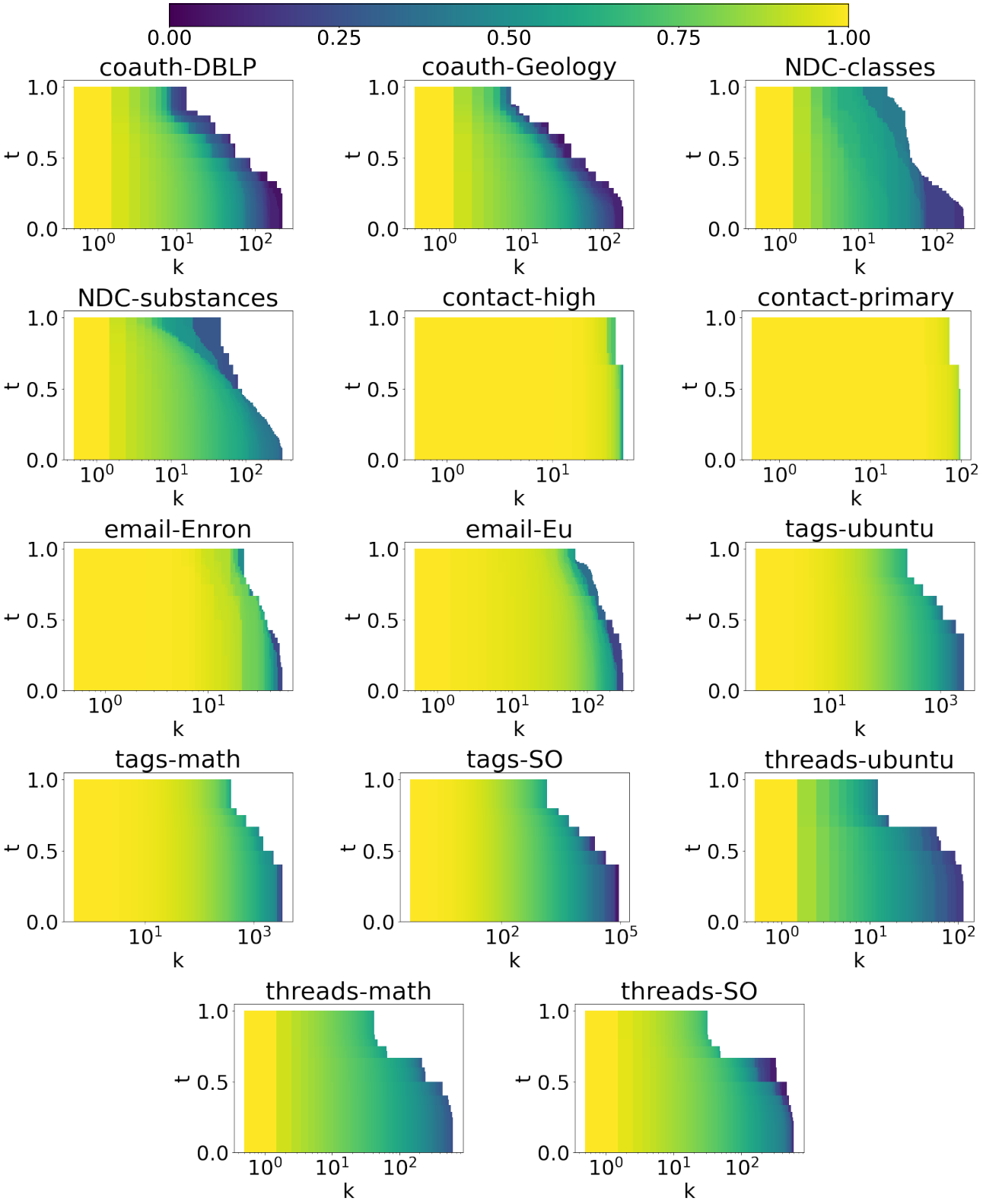


Fig. 2: (k, t)-hypercore sizes.

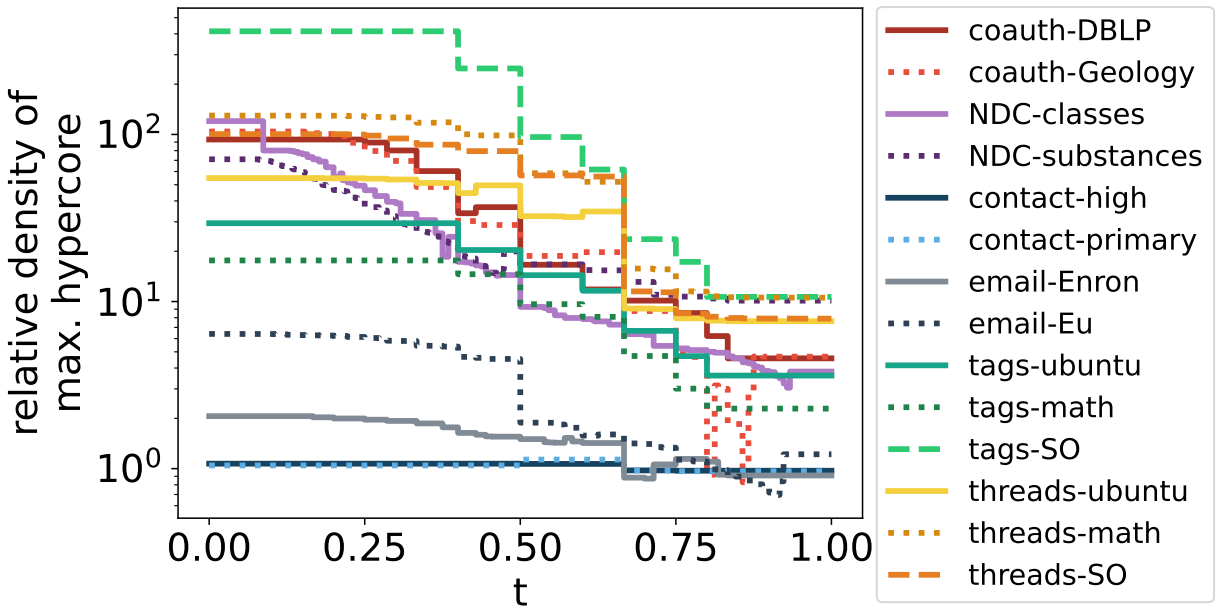


Fig. 3: **Overall, hypercores are much denser than the whole hypergraph, and the density decreases as t increases.** For each dataset, we report the relative density of the (c_t^*, t)-hypercore (i.e., the (k, t)-hypercore with maximal k) w.r.t t.

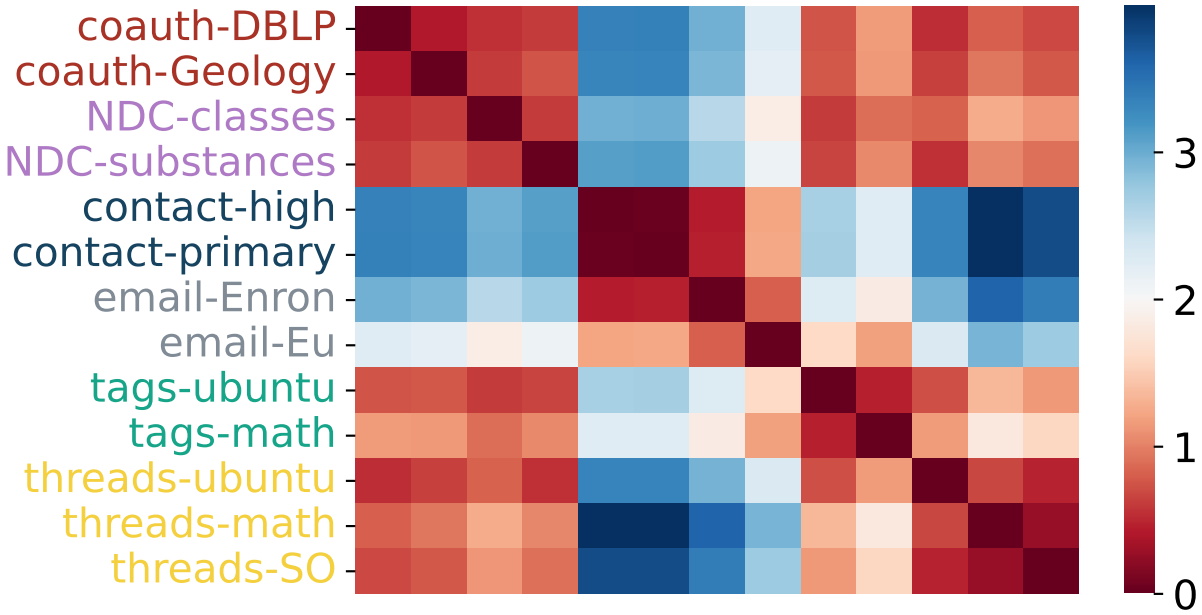
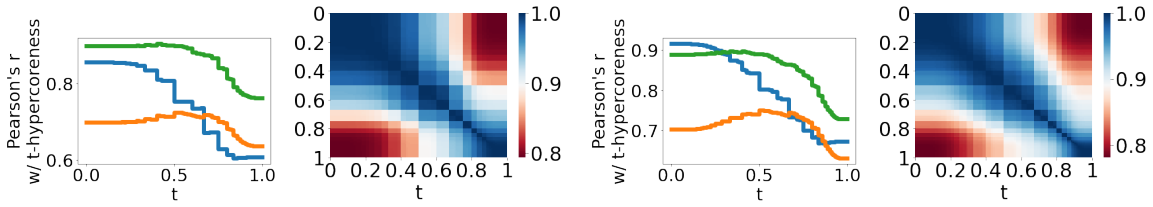
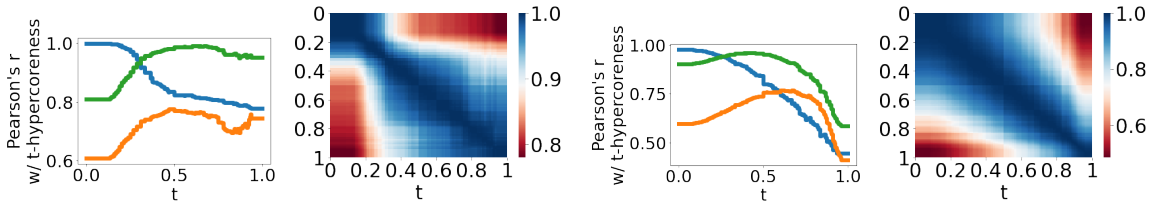


Fig. 4: **Datasets in the same domain have similar patterns of relative density.** We show the RDMD distance (Definition 3) between each pair of datasets except for the tag-SO dataset. Overall, the RDMD distance is small between hypergraphs in the same domain. Specifically, the average distance is 1.741 overall and 0.456 within domains. The two mean values are significantly different with 0.0035 as the p-value of the t-test.



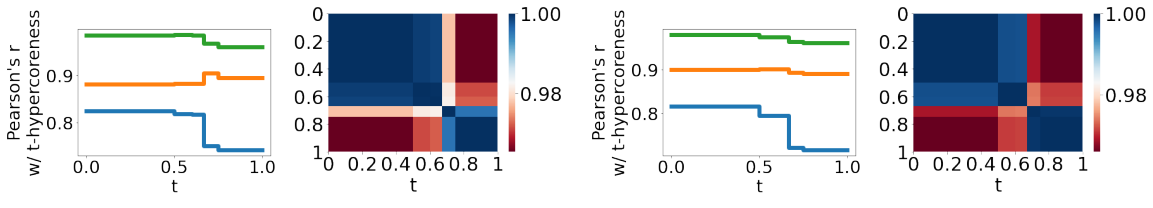
(a) coauth-DBLP

(b) coauth-Geology



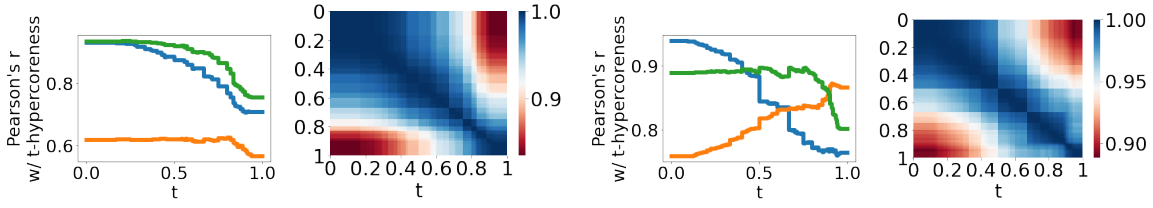
(c) NDC-classes

(d) NDC-substances



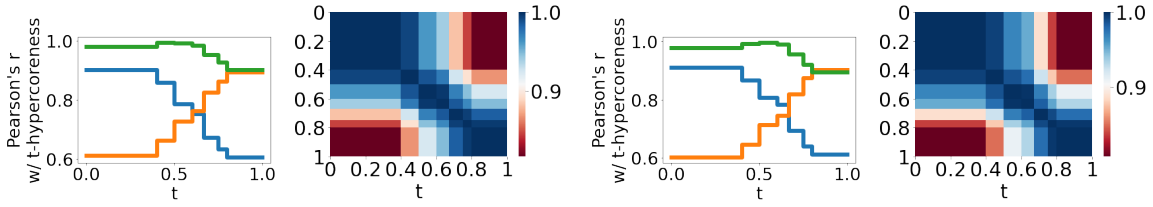
(e) contact-high

(f) contact-primary



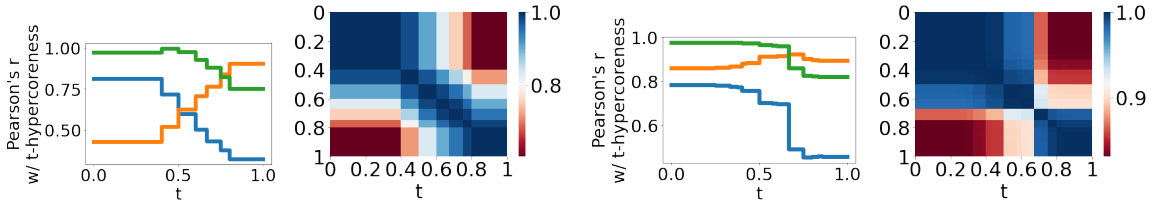
(g) email-Enron

(h) email-Eu



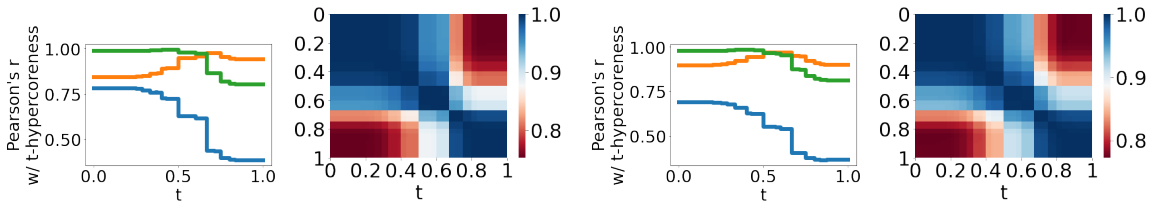
(i) tags-ubuntu

(j) tags-math



(k) tags-SO

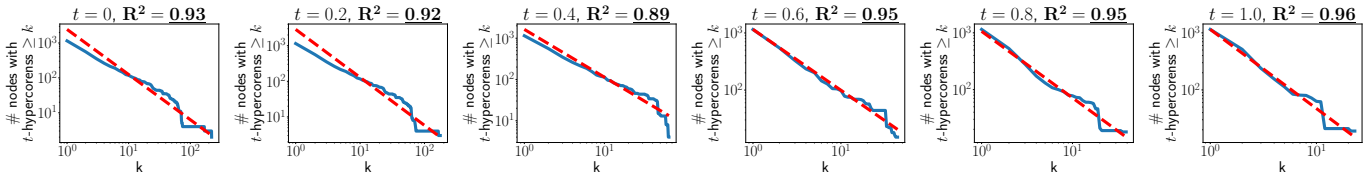
(l) threads-ubuntu



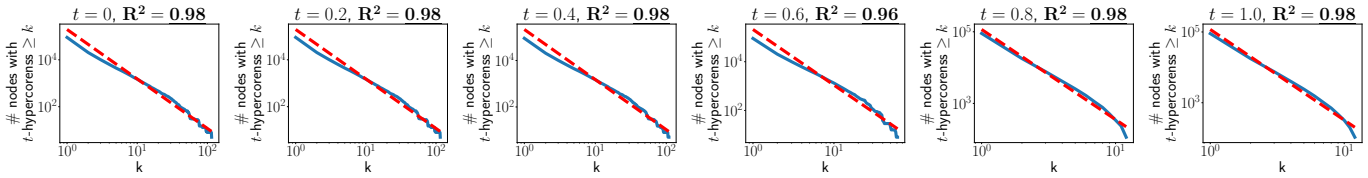
(m) threads-math

(n) threads-SO

Fig. 5: Correlations.

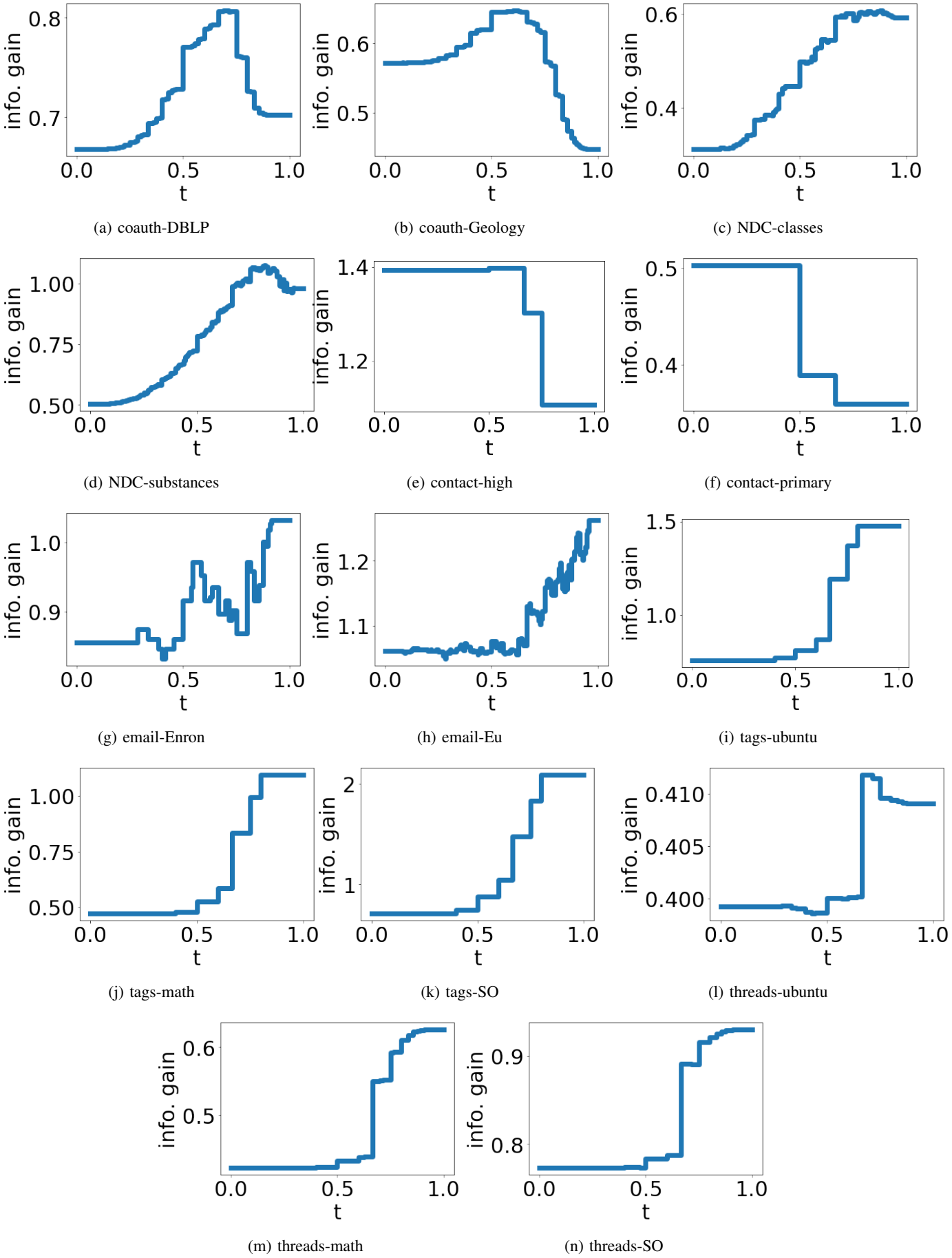


(a) NDC-classes



(b) threads-ubuntu

Fig. 6: **t-Hypercoreness consistently follows power-law distributions in some datasets.** For the NDC-classes and threads-ubuntu datasets with $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, we show the numbers of nodes with t-hypercoreness at least k with different k values. Each red dashed line represents the result of power-law fitting, i.e., the linear regression in log-log scale, with the R^2 value above each subfigure. In the two dataset, t-hypercoreness consistently and strongly follows a power law.



(a) coauth-DBLP (b) coauth-Geology (c) NDC-classes

(d) NDC-substances (e) contact-high (f) contact-primary

(g) email-Enron (h) email-Eu (i) tags-ubuntu

(j) tags-math (k) tags-SO (l) threads-ubuntu

(m) threads-math (n) threads-SO

Fig. 7: Information gain.

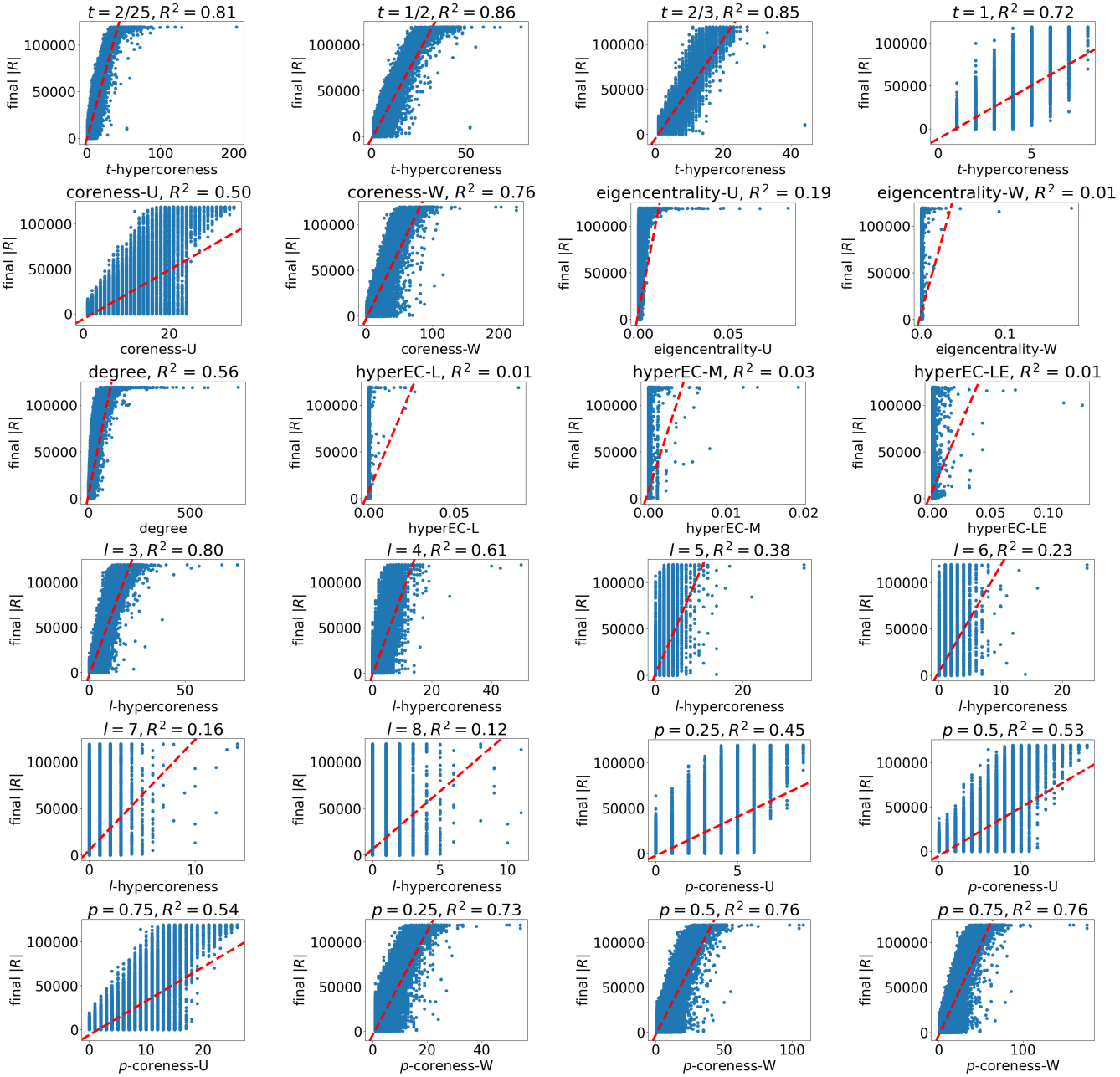


Fig. 8: Influence: coauth-DBLP ($\beta = 0.05$, $\gamma = 1.0$, 10% nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

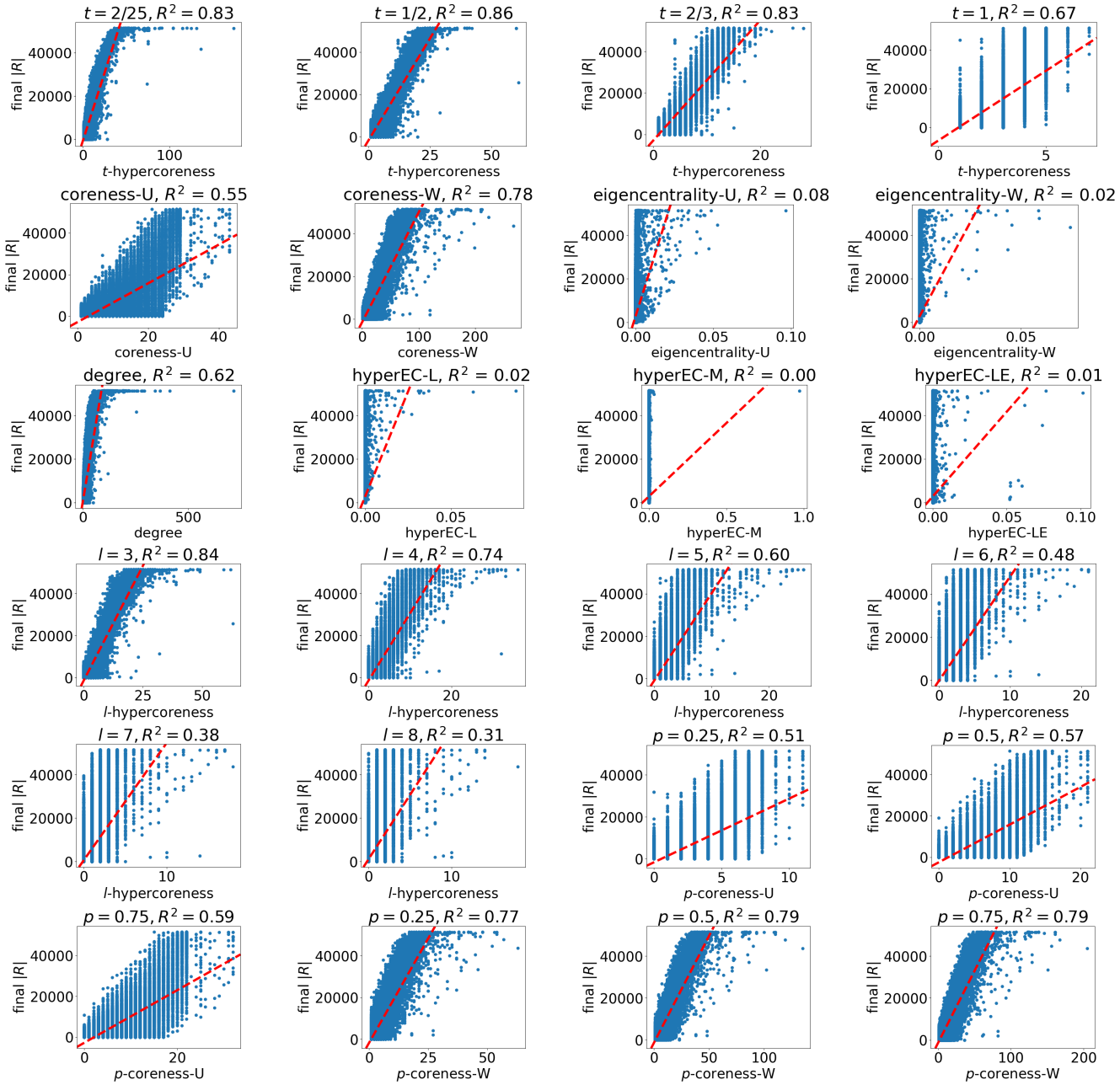


Fig. 9: Influence: coauth-Geology ($\beta = 0.05$, $\gamma = 1.0$, 10% nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

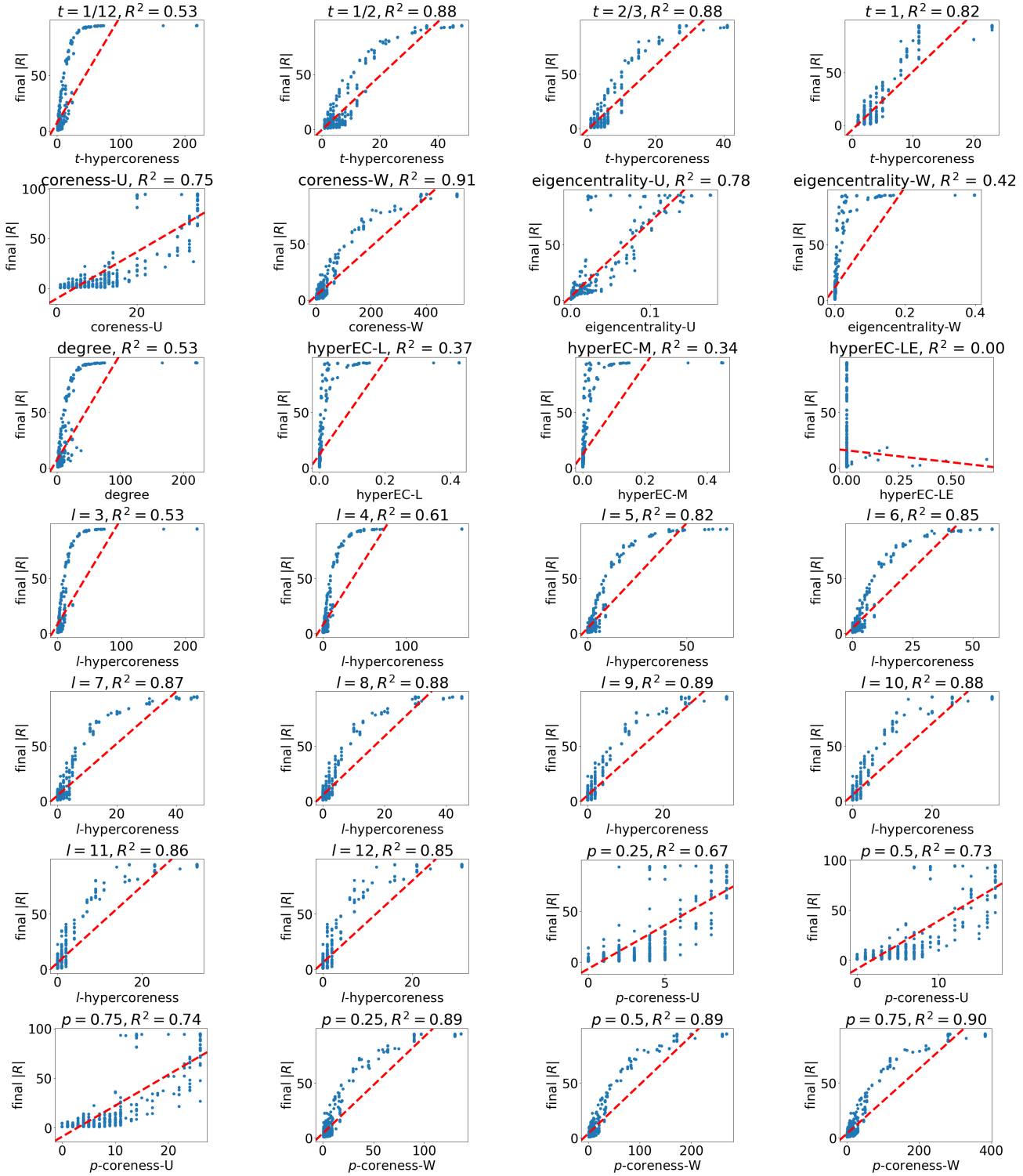


Fig. 10: Influence: NDC-classes ($\beta = 0.05$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

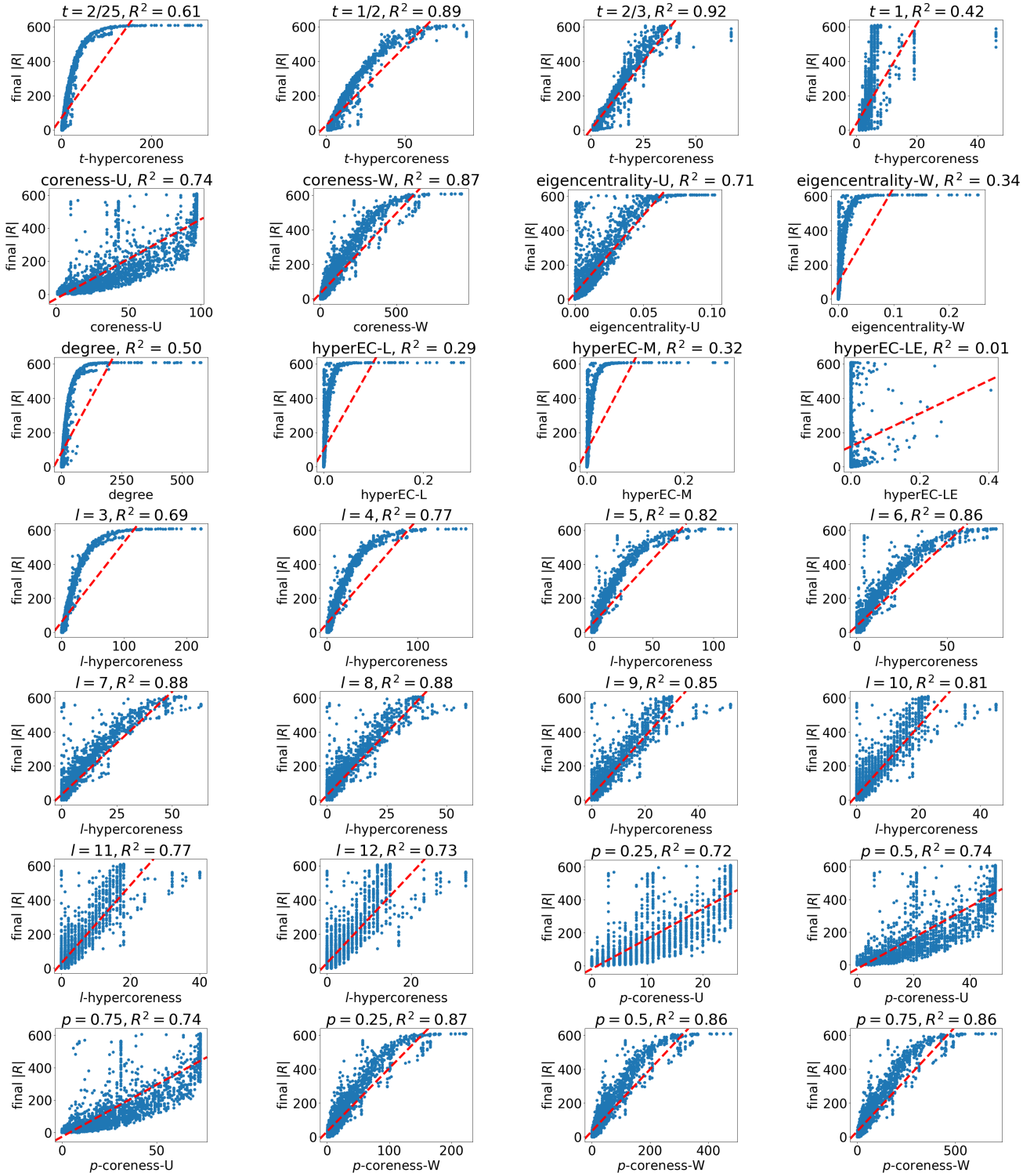


Fig. 11: Influence: NDC-substances ($\beta = 0.025$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

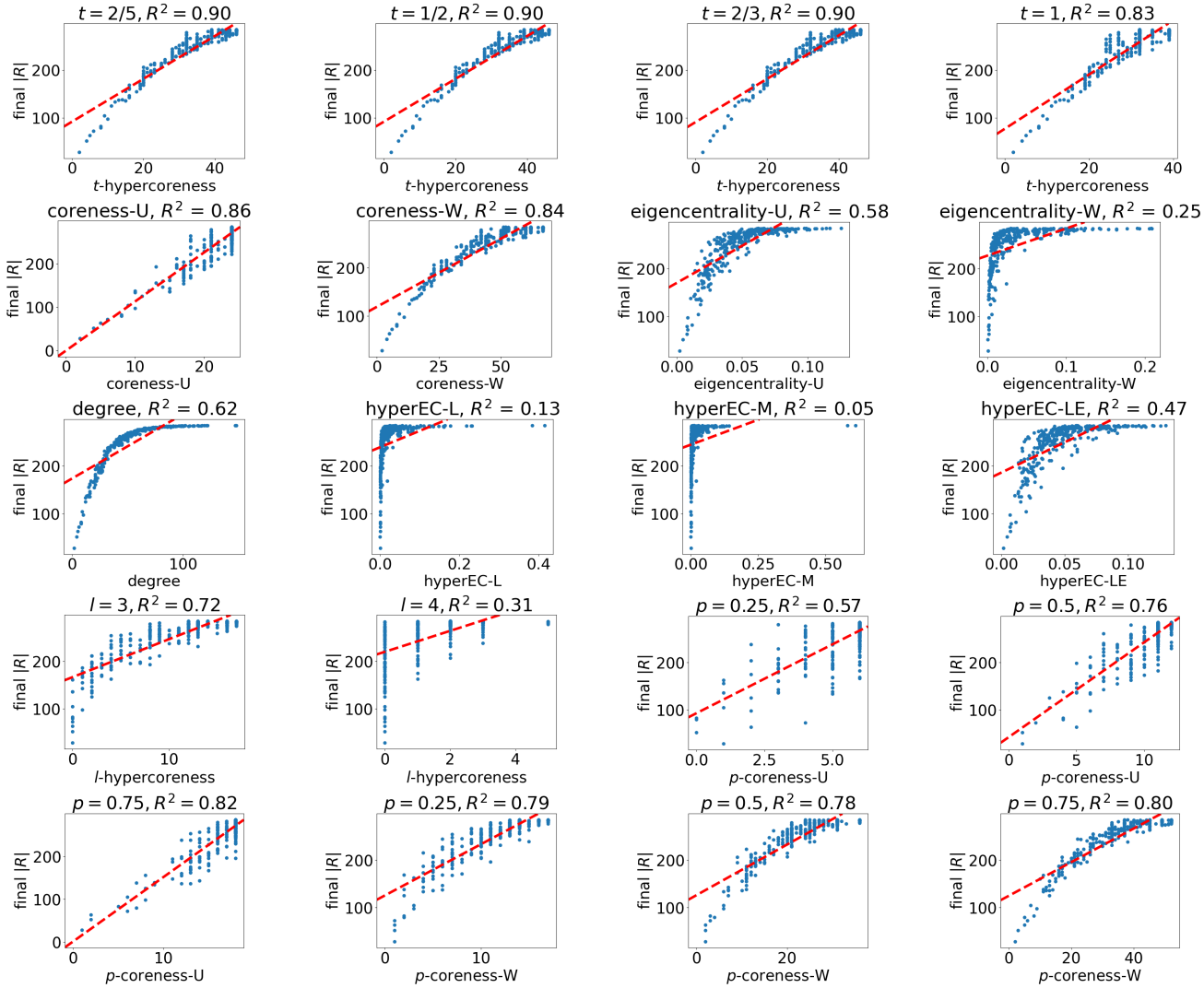


Fig. 12: Influence: contact-high ($\beta = 0.05$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

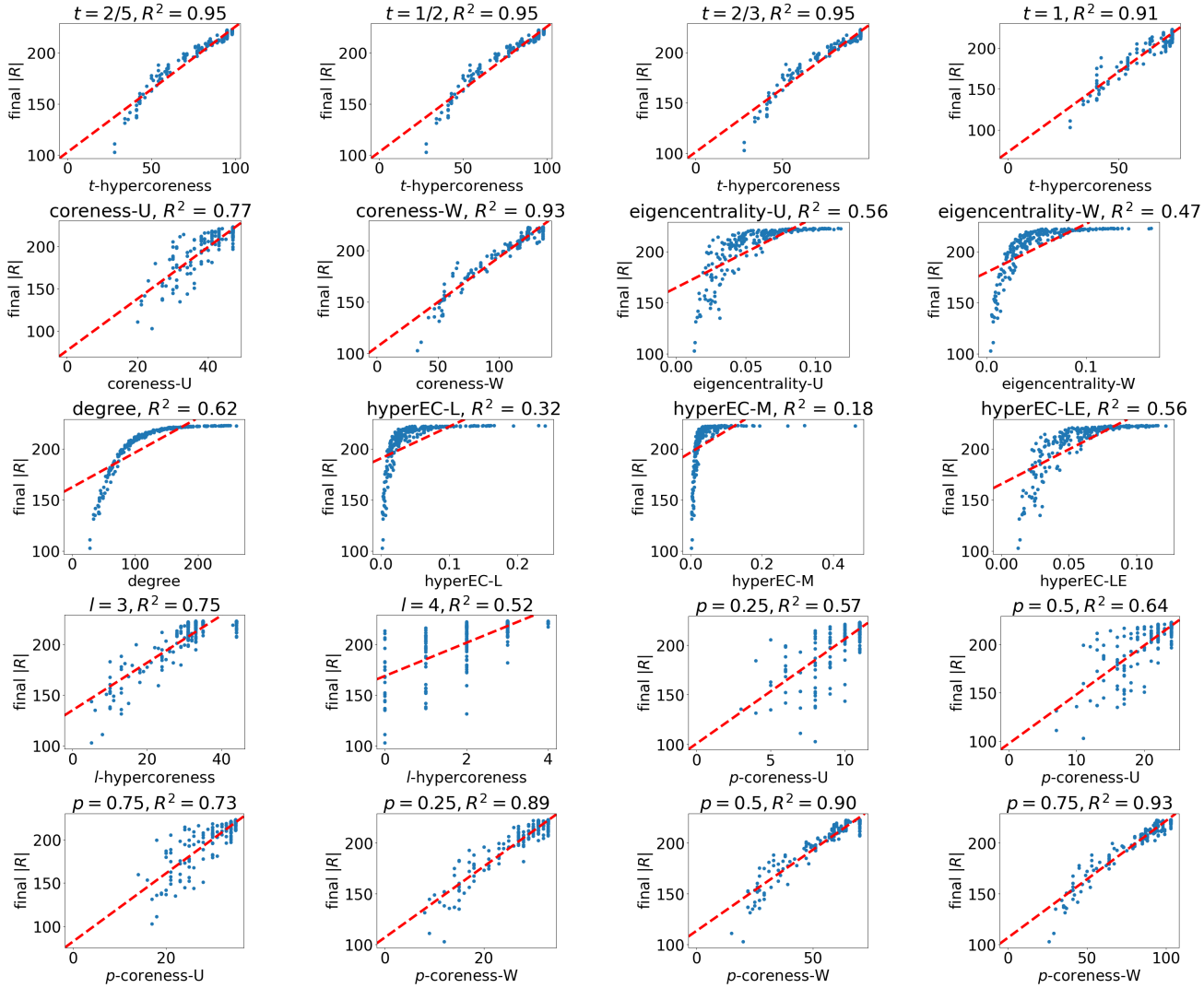


Fig. 13: Influence: contact-primary ($\beta = 0.025$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

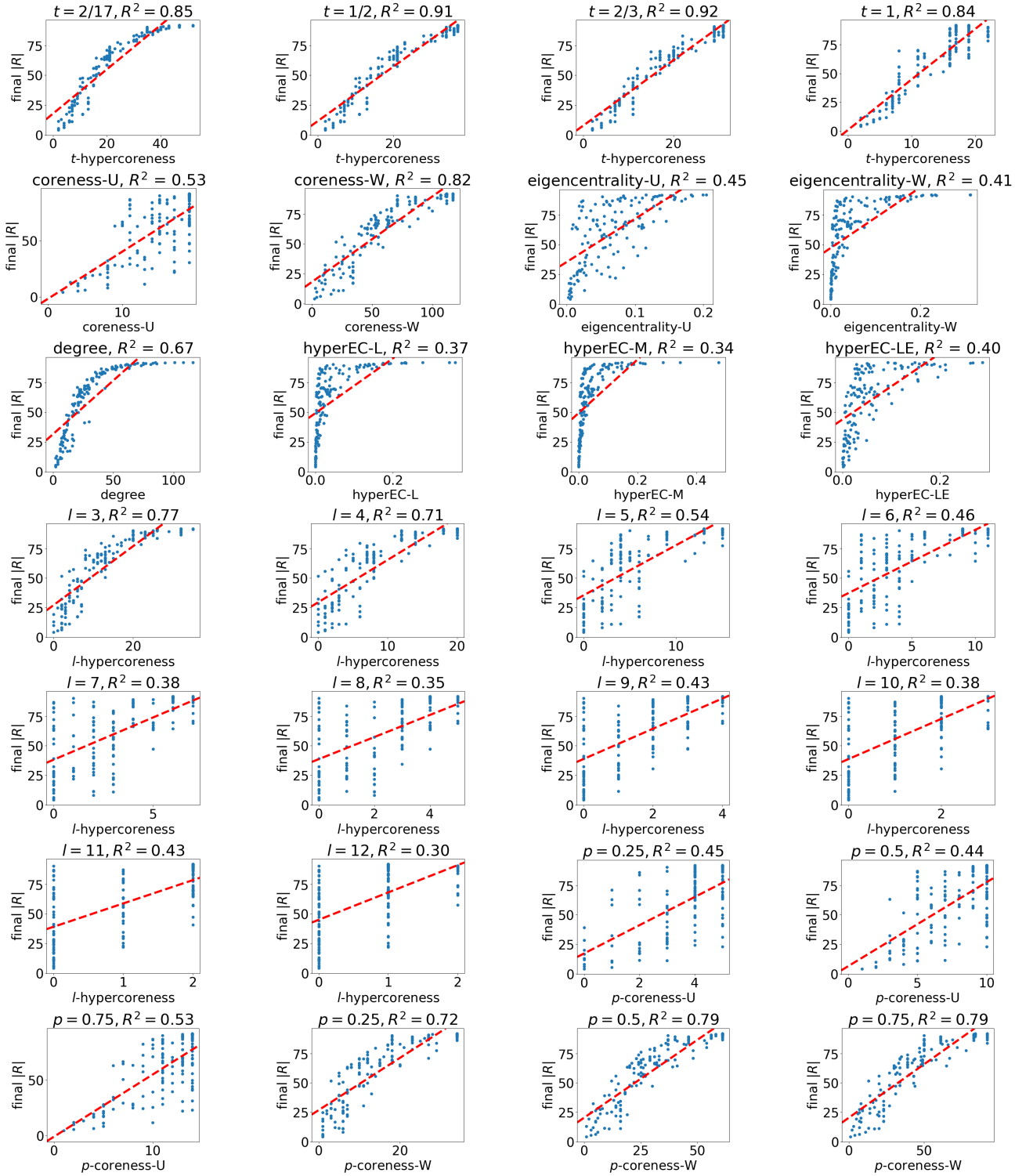


Fig. 14: Influence: email-Enron ($\beta = 0.05$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

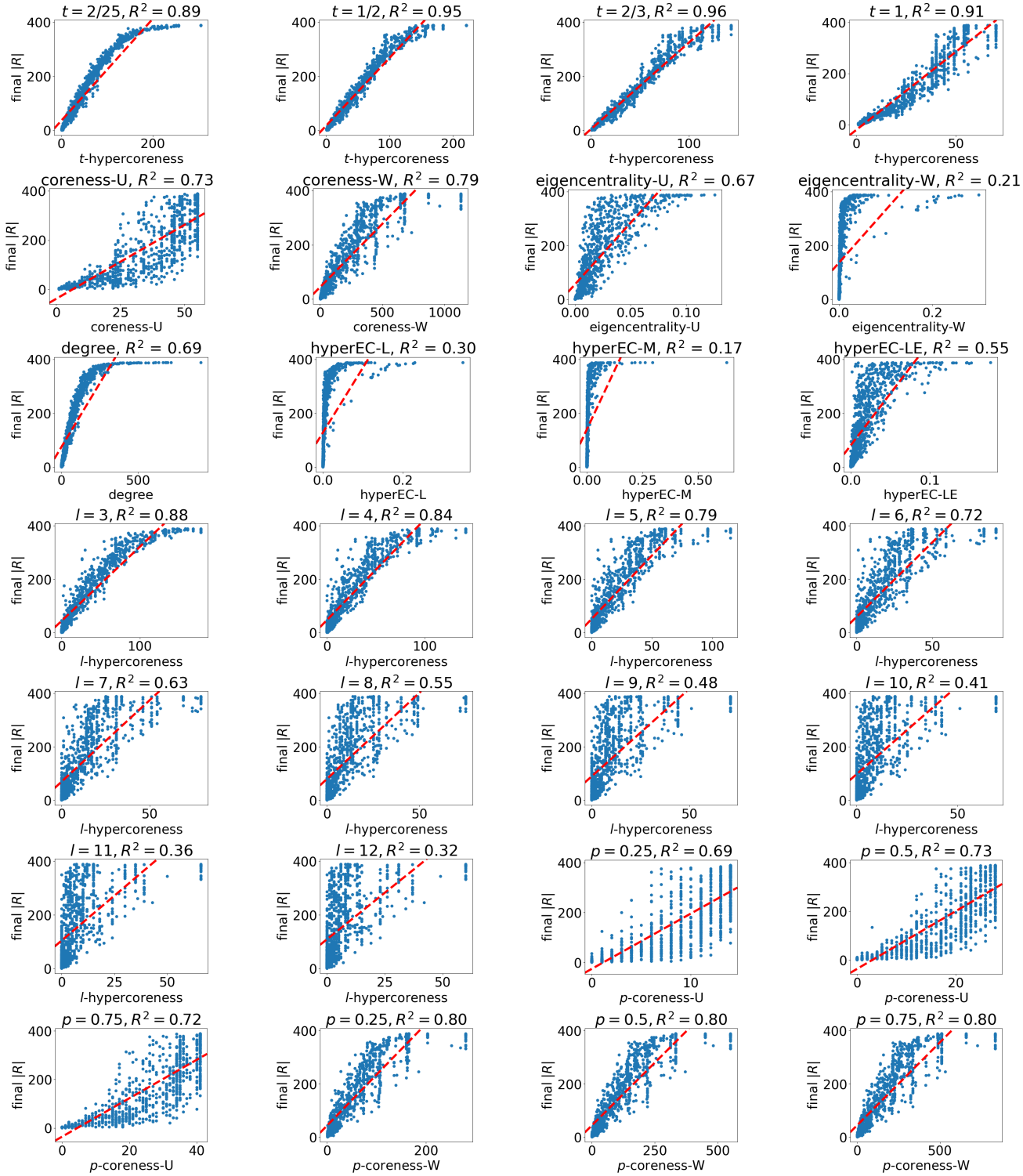


Fig. 15: Influence: email-Eu ($\beta = 0.01$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

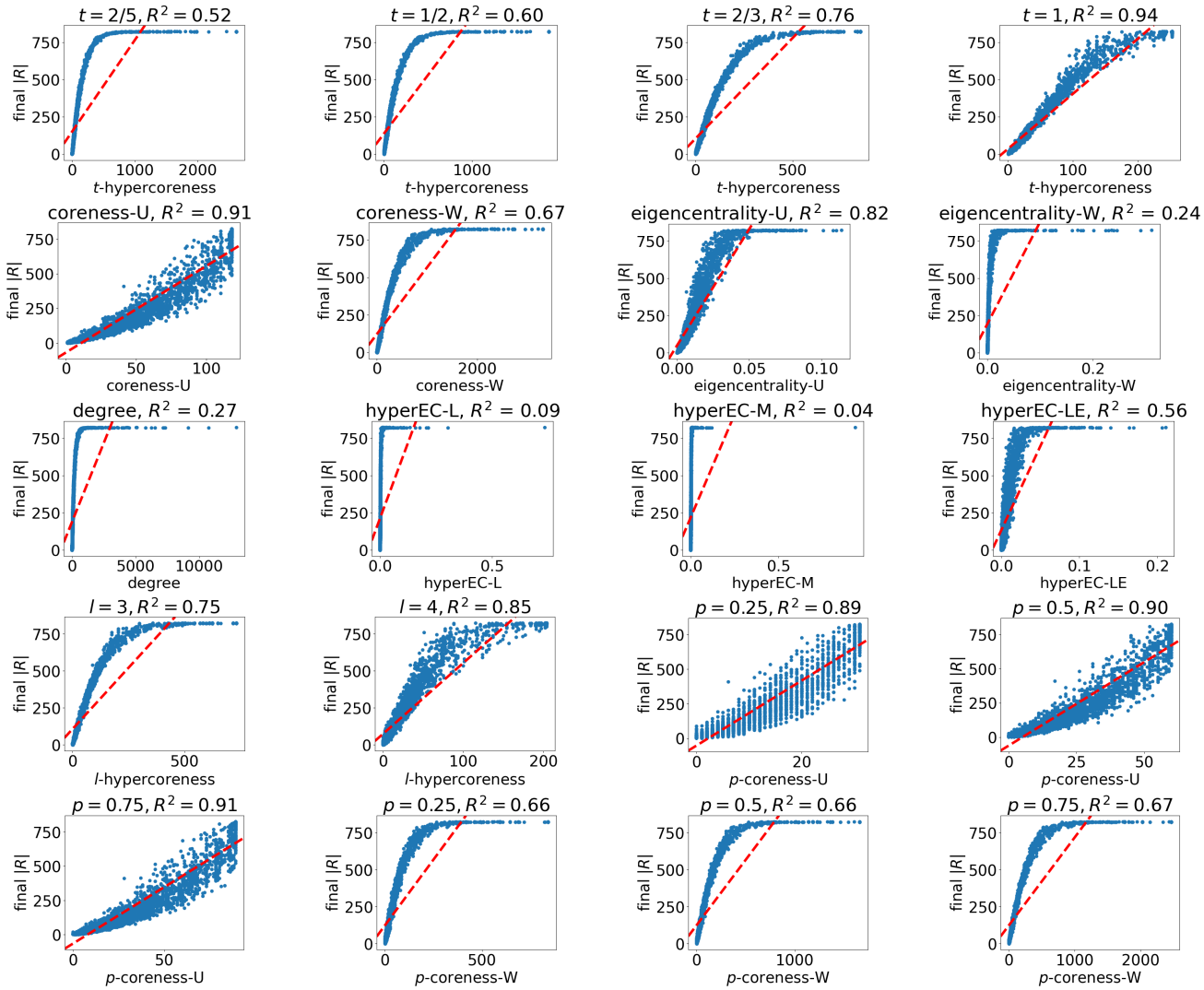


Fig. 16: Influence: tags-ubuntu ($\beta = 0.01$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

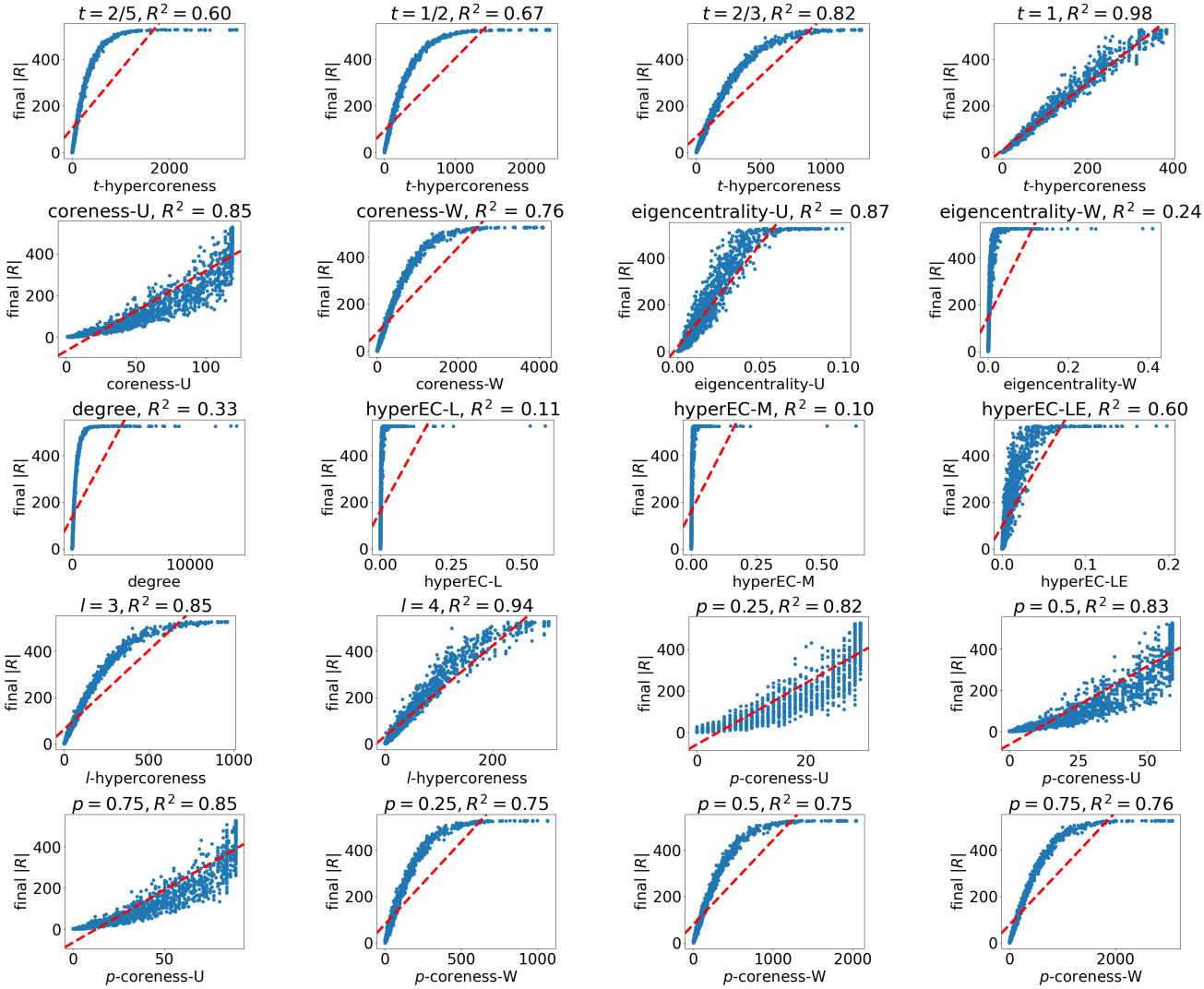


Fig. 17: Influence: tags-math ($\beta = 0.0025$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

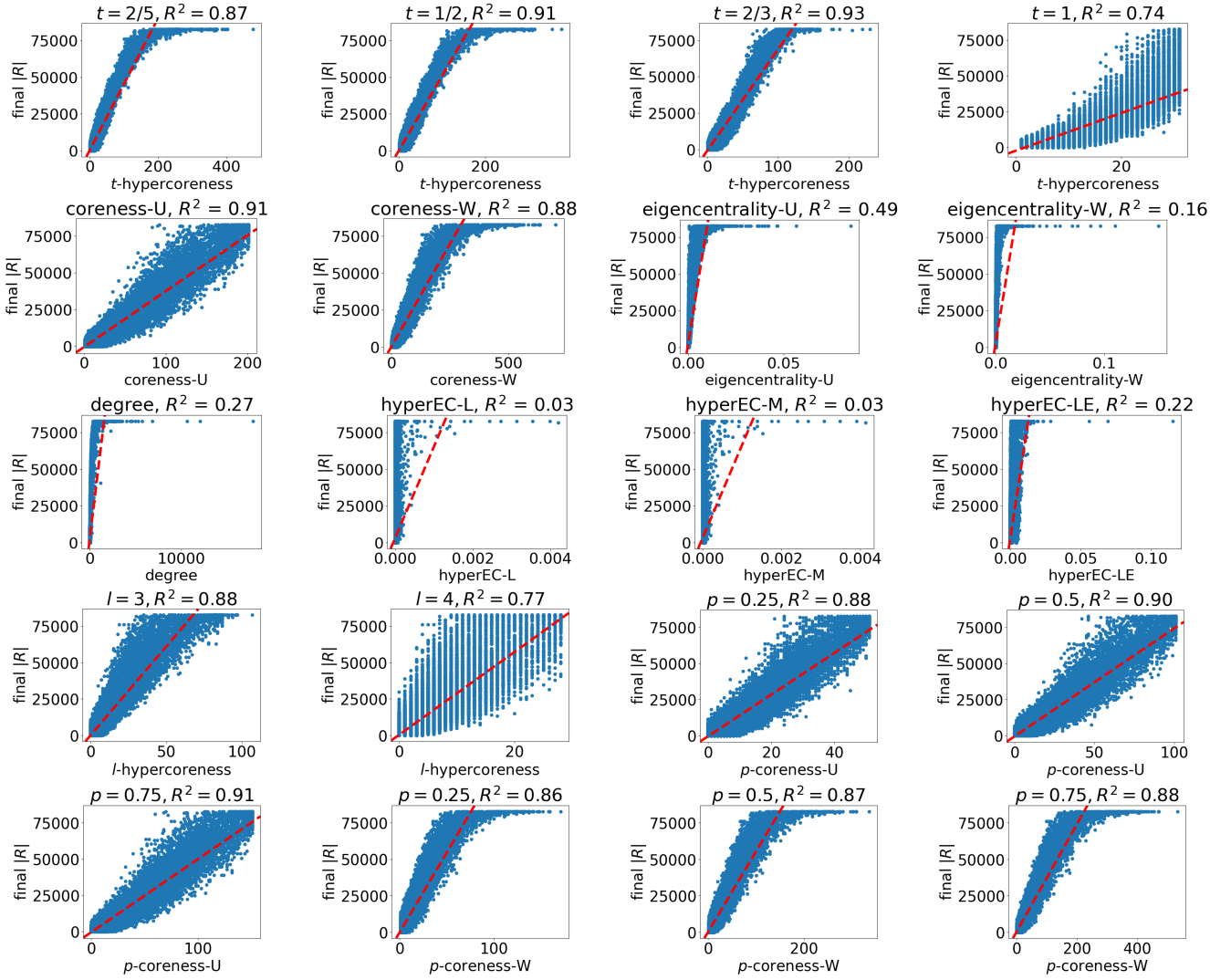


Fig. 18: Influence: tags-SO ($\beta = 0.0025$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

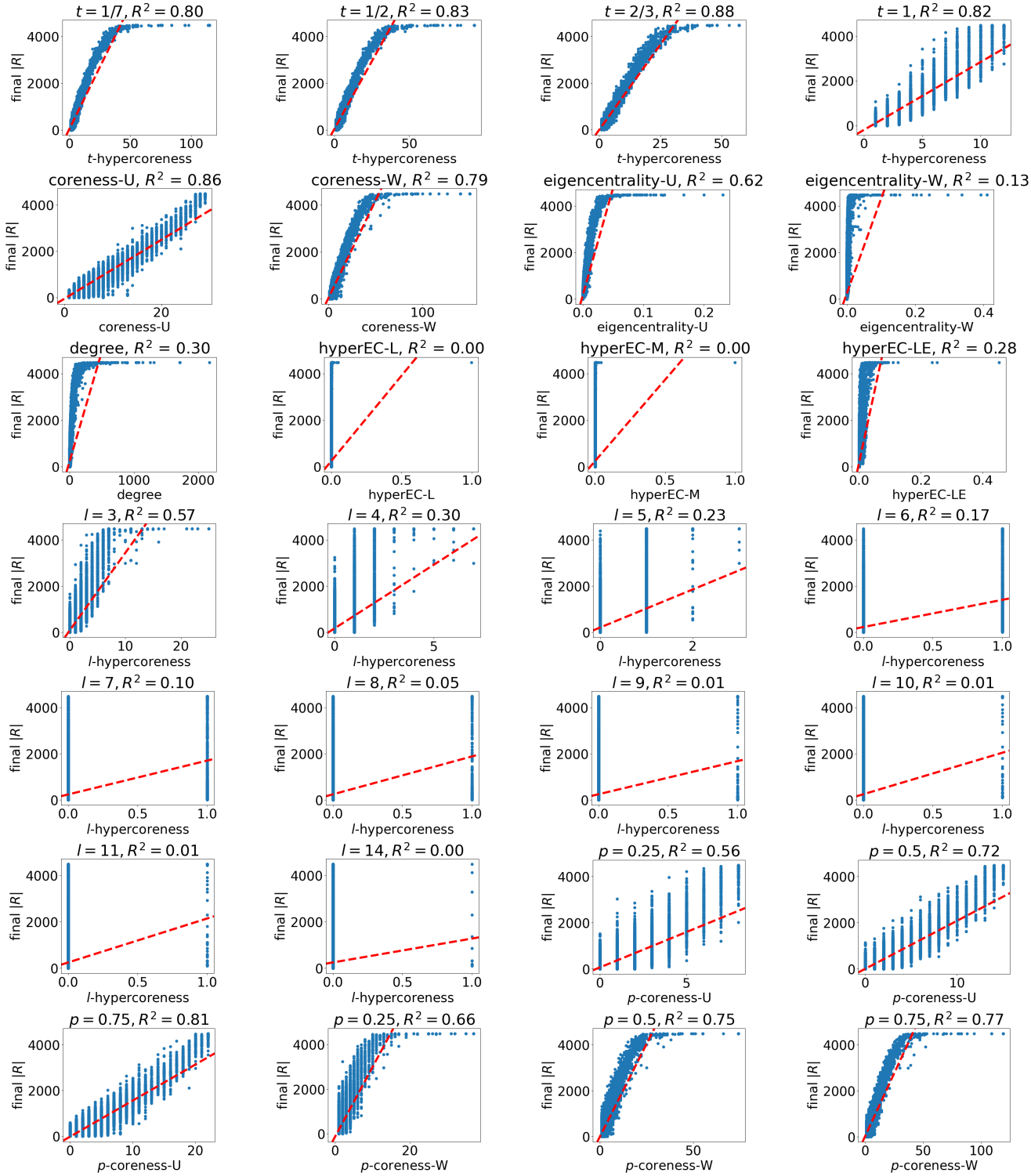


Fig. 19: Influence: threads-ubuntu ($\beta = 0.05$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

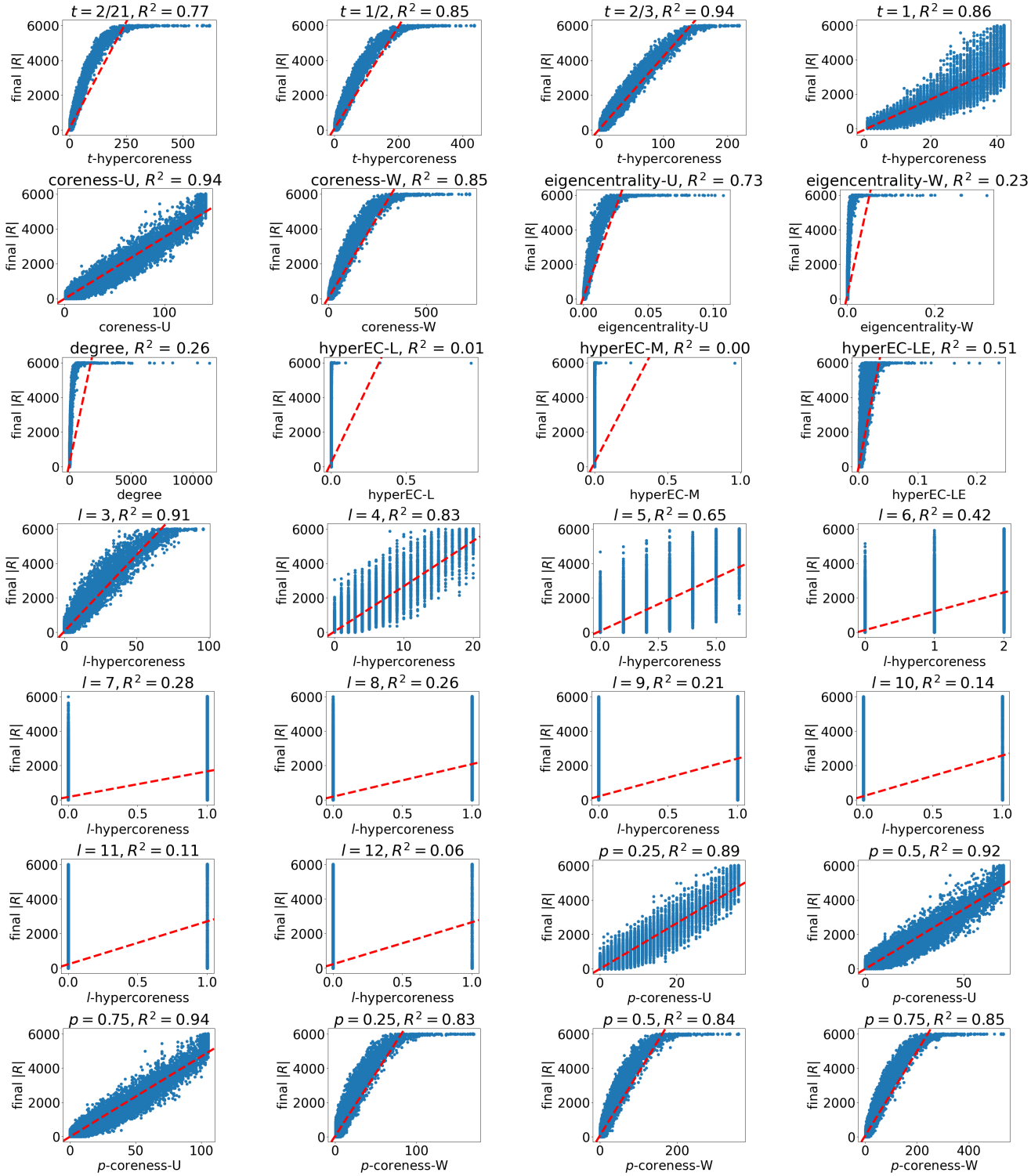


Fig. 20: Influence: threads-math ($\beta = 0.01$, $\gamma = 1.0$, all nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

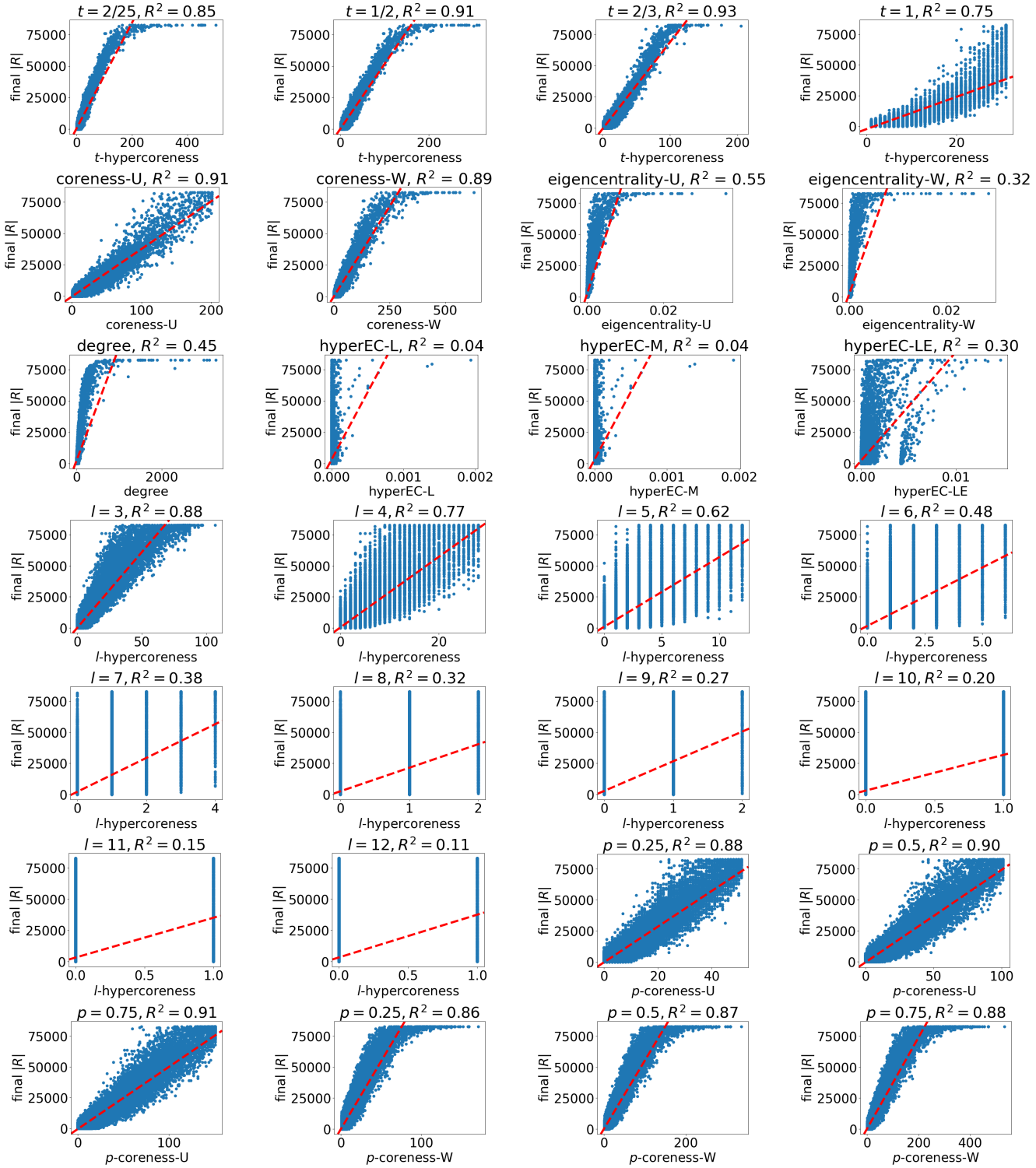


Fig. 21: Influence: threads-SO ($\beta = 0.01$, $\gamma = 1.0$, 10% nodes). The $(l = 2)$-hypercoreness is included in our proposed concept as the $(t = 0)$-hypercoreness, which is the first subfigure with minimum t.

TABLE VII: Full results of the collapsed (k, t)-hypercore problem ($b = 100$). Left: running time (in seconds); Right: reduction in the hypercore size.

dataset, k, t	HYPERCKC	HYCOM-1	HYCOM-10	HYCOM-100	HYCOM+
coauth-DBLP, 5, 0	63.01 / 1,176	15.80 / 680	16.44 / 729	17.50 / 812	5.37 / 1,088
coauth-DBLP, 5, 0.2	63.38 / 1,185	15.87 / 691	16.47 / 738	17.54 / 819	5.41 / 1,091
coauth-DBLP, 5, 0.4	70.43 / 1,378	18.21 / 831	18.87 / 897	19.91 / 986	6.28 / 1,250
coauth-DBLP, 5, 0.6	113.38 / 2,536	34.17 / 1,675	35.00 / 1,795	36.18 / 1,969	11.83 / 2,259
coauth-DBLP, 5, 0.8	130.19 / 6,681	43.21 / 4,487	43.90 / 5,075	45.86 / 5,785	13.88 / 5,943
coauth-DBLP, 5, 1	87.75 / 7,523	25.40 / 5,914	26.10 / 6,108	27.93 / 6,752	8.70 / 6,731
coauth-DBLP, 10, 0	50.42 / 1,181	9.74 / 661	10.08 / 699	11.37 / 832	2.51 / 1,008
coauth-DBLP, 10, 0.2	50.82 / 1,195	9.74 / 676	10.21 / 716	11.46 / 852	2.55 / 1,019
coauth-DBLP, 10, 0.4	56.94 / 1,462	11.38 / 844	11.78 / 877	13.13 / 1,001	3.08 / 1,206
coauth-DBLP, 10, 0.6	75.03 / 2,798	16.05 / 1,621	16.45 / 1,711	18.14 / 2,050	4.30 / 2,328
coauth-DBLP, 10, 0.8	0.24 / 2,699	0.034 / 2,742	0.051 / 2,699	0.13 / 2,699	0.13 / 2,740
coauth-DBLP, 20, 0	33.32 / 1,251	5.21 / 590	5.45 / 632	6.72 / 788	1.02 / 945
coauth-DBLP, 20, 0.2	33.56 / 1,286	5.24 / 588	5.50 / 617	6.73 / 802	1.03 / 1,000
coauth-DBLP, 20, 0.4	34.72 / 1,558	5.14 / 749	5.41 / 784	6.71 / 1,003	1.11 / 1,206
coauth-DBLP, 20, 0.6	13.67 / 2,817	2.06 / 1,813	2.19 / 1,975	3.17 / 2,288	0.51 / 2,218
coauth-Geology, 5, 0	16.65 / 859	3.83 / 343	4.03 / 352	4.44 / 423	1.35 / 708
coauth-Geology, 5, 0.2	16.79 / 877	3.84 / 344	4.05 / 352	4.46 / 429	1.36 / 720
coauth-Geology, 5, 0.4	20.55 / 1,120	4.60 / 386	4.86 / 403	5.26 / 518	1.77 / 865
coauth-Geology, 5, 0.6	36.58 / 2,004	9.22 / 660	9.58 / 686	10.07 / 914	3.29 / 1,578
coauth-Geology, 5, 0.8	32.32 / 5,334	7.58 / 2,992	7.92 / 3,164	8.45 / 3,986	2.53 / 3,891
coauth-Geology, 5, 1	4.67 / 7,294	0.94 / 9,498	0.94 / 9,576	1.24 / 9,536	0.47 / 6,891
coauth-Geology, 10, 0	12.81 / 856	2.65 / 316	2.77 / 328	3.14 / 423	0.64 / 720
coauth-Geology, 10, 0.2	12.97 / 857	2.68 / 331	2.80 / 349	3.17 / 441	0.65 / 715
coauth-Geology, 10, 0.4	16.81 / 1,132	3.34 / 374	3.49 / 386	3.89 / 524	0.85 / 926
coauth-Geology, 10, 0.6	21.51 / 2,652	3.92 / 914	4.03 / 966	4.61 / 1,311	1.03 / 1,867
coauth-Geology, 10, 0.8	0.0027 / 299	0.0023 / 388	0.0027 / 299	0.0027 / 299	0.03 / 388
coauth-Geology, 20, 0	9.08 / 940	1.73 / 319	1.81 / 330	2.14 / 462	0.29 / 689
coauth-Geology, 20, 0.2	9.18 / 977	1.74 / 334	1.81 / 343	2.22 / 476	0.30 / 713
coauth-Geology, 20, 0.4	10.13 / 1,470	1.78 / 528	1.85 / 550	2.23 / 742	0.32 / 1,105
coauth-Geology, 20, 0.6	0.031 / 459	0.0093 / 529	0.014 / 459	0.030 / 459	0.033 / 527
tags-SO, 5, 0	191.84 / 1,042	42.03 / 930	55.66 / 962	108.93 / 1,012	7.87 / 1,042
tags-SO, 5, 0.6	361.12 / 2,103	111.72 / 1,924	125.07 / 1,968	184.97 / 2,056	18.83 / 2,102
tags-SO, 5, 0.8	605.38 / 9,669	185.49 / 9,232	208.27 / 9,306	296.96 / 9,550	25.39 / 9,663
tags-SO, 5, 1	488.02 / 15,560	152.06 / 15,057	177.59 / 15,150	252.22 / 15,517	20.05 / 15,552
tags-SO, 10, 0	202.64 / 1,258	42.03 / 1,126	55.65 / 1,160	107.56 / 1,203	8.26 / 1,256
tags-SO, 10, 0.6	380.46 / 2,638	110.93 / 2,423	124.09 / 2,450	185.16 / 2,609	18.99 / 2,640
tags-SO, 10, 0.8	609.48 / 11,253	179.73 / 10,919	203.65 / 11,035	291.75 / 11,227	24.31 / 11,244
tags-SO, 10, 1	476.07 / 16,950	145.35 / 16,687	169.36 / 16,750	240.98 / 16,947	19.44 / 16,951
tags-SO, 20, 0.4	213.17 / 1,328	41.38 / 1,198	54.94 / 1,224	107.45 / 1,313	8.49 / 1,327
tags-SO, 20, 0.6	394.77 / 2,905	108.75 / 2,743	122.387 / 2,796	183.53 / 2,885	18.99 / 2,900
tags-SO, 20, 0.8	586.81 / 11,431	166.97 / 11,147	190.06 / 11,254	275.55 / 11,439	22.98 / 11,438
tags-SO, 20, 1	445.40 / 16,427	131.03 / 16,302	153.92 / 16,355	219.95 / 16,427	18.12 / 16,437
threads-math, 5, 0	18.65 / 4,549	2.89 / 4,319	3.61 / 4,395	7.04 / 4,543	0.70 / 4,552
threads-math, 5, 0.4	18.71 / 4,564	2.92 / 4,345	3.62 / 4,419	7.12 / 4,557	0.70 / 4,567
threads-math, 5, 0.6	21.23 / 5,339	3.93 / 5,209	4.62 / 5,259	8.35 / 5,339	0.91 / 5,336
threads-math, 5, 0.8	32.89 / 9,366	8.31 / 9,262	9.52 / 9,296	14.53 / 9,366	1.72 / 9,359
threads-math, 5, 1	32.61 / 9,741	8.59 / 9,619	9.74 / 9,670	14.54 / 9,741	1.72 / 9,731
threads-math, 10, 0	15.93 / 2,708	2.40 / 2,545	3.02 / 2,595	6.33 / 2,697	0.50 / 2,708
threads-math, 10, 0.4	16.08 / 2,722	2.45 / 2,565	3.07 / 2,615	6.39 / 2,713	0.51 / 2,708
threads-math, 10, 0.6	18.93 / 3,318	3.26 / 3,228	3.96 / 3,241	7.65 / 3.316	0.67 / 3,320
threads-math, 10, 0.8	22.86 / 5,618	4.78 / 5,522	5.71 / 5,582	9.51 / 5,618	0.94 / 5,608
threads-math, 10, 1	21.49 / 5,773	4.53 / 5,644	5.32 / 5,707	8.96 / 5,768	0.87 / 5,760
threads-math, 20, 0	13.06 / 1,658	1.95 / 1,573	2.49 / 1,598	5.59 / 1,661	0.37 / 1,673
threads-math, 20, 0.4	13.25 / 1,691	1.99 / 1,594	2.55 / 1,623	5.64 / 1,690	0.38 / 1,683
threads-math, 20, 0.6	16.12 / 2,157	2.62 / 2,099	3.29 / 2,143	6.69 / 2,160	0.49 / 2,159
threads-math, 20, 0.8	12.80 / 3,693	2.09 / 3,627	2.63 / 3,667	5.03 / 3,692	0.43 / 3,701
threads-math, 20, 1	10.83 / 3,625	1.68 / 3,589	2.12 / 3,616	4.25 / 3,635	0.36 / 3,622
threads-SO, 5, 0	634.38 / 13,739	183.04 / 11,927	186.79 / 12,209	205.62 / 13,083	41.64 / 13,736
threads-SO, 5, 0.4	642.64 / 13,748	182.99 / 11,935	187.02 / 12,216	205.41 / 13,093	41.60 / 13,745
threads-SO, 5, 0.6	653.52 / 15,050	224.97 / 13,462	227.72 / 13,659	240.00 / 14,536	49.05 / 15,046
threads-SO, 5, 0.8	1409.66 / 31,383	522.59 / 29,317	529.38 / 30,198	565.72 / 31,133	112.59 / 31,336
threads-SO, 5, 1	1592.06 / 36,082	572.42 / 34,044	580.25 / 34,663	638.44 / 35,913	133.73 / 36,013
threads-SO, 10, 0	562.91 / 8,425	148.98 / 7,432	154.16 / 7,532	170.28 / 8,100	29.87 / 8,417
threads-SO, 10, 0.4	564.14 / 8,449	149.99 / 7,456	153.27 / 7,558	170.84 / 8,124	30.08 / 8,433
threads-SO, 10, 0.6	641.40 / 10,100	204.75 / 9,210	205.99 / 9,331	219.53 / 9,879	38.35 / 10,065
threads-SO, 10, 0.8	1105.59 / 21,202	340.25 / 19,490	346.77 / 19,819	375.53 / 21,091	67.12 / 21,099
threads-SO, 10, 1	1131.47 / 23,234	342.87 / 22,056	349.27 / 22,311	382.53 / 23,058	66.88 / 23,115
threads-SO, 20, 0	477.38 / 5,155	120.34 / 4,502	123.52 / 4,621	140.99 / 4,943	19.31 / 5,124
threads-SO, 20, 0.4	482.70 / 5,177	123.59 / 4,539	128.87 / 4,645	145.46 / 4,967	19.66 / 5,154
threads-SO, 20, 0.6	610.01 / 7,060	165.86 / 6,457	170.97 / 6,591	187.87 / 6,889	28.21 / 7,019
threads-SO, 20, 0.8	690.80 / 14,129	158.07 / 13,622	161.77 / 13,725	185.95 / 14,029	27.34 / 14,009
threads-SO, 20, 1	533.83 / 14,734	109.77 / 14,259	113.39 / 14,358	134.04 / 14,614	20.65 / 14,461

TABLE VIII: The basic statistics of the 6 additional datasets.

Dataset	$\|V\|$	$\|E\|$	max./avg. $d(v)$	max./avg. $\|e\|$
linux-kernal-mailing	15,950	122,531	9,562/22.44	25/2.92
marvel	8,448	4,017	30/3.03	25/6.38
foursquare-NYC-restaurant	2,008	2,242	76/5.90	25/5.29
foursquare-NYC	1,083	13,809	489/53.00	25/4.16
foursquare-Tokyo	2,293	24,703	380/44.63	25/4.14
wikinews	11,413	97,833	24,002/32.42	25/3.78

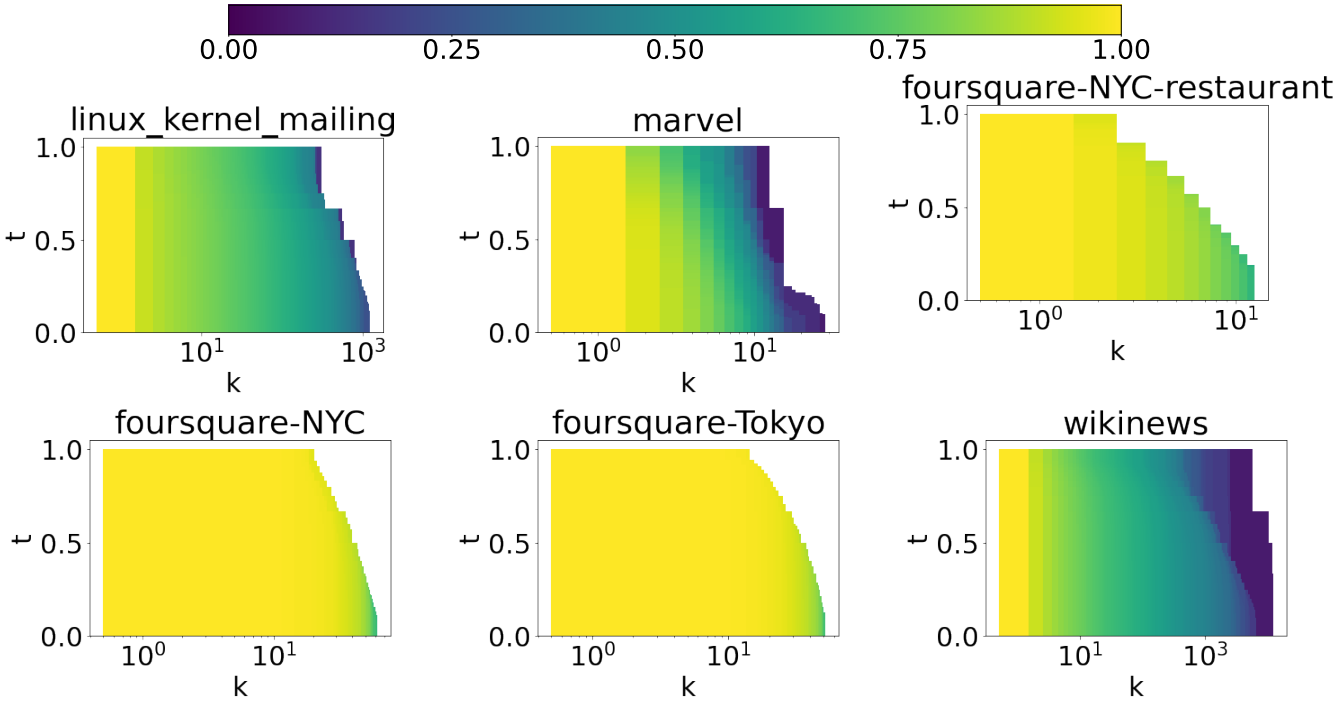
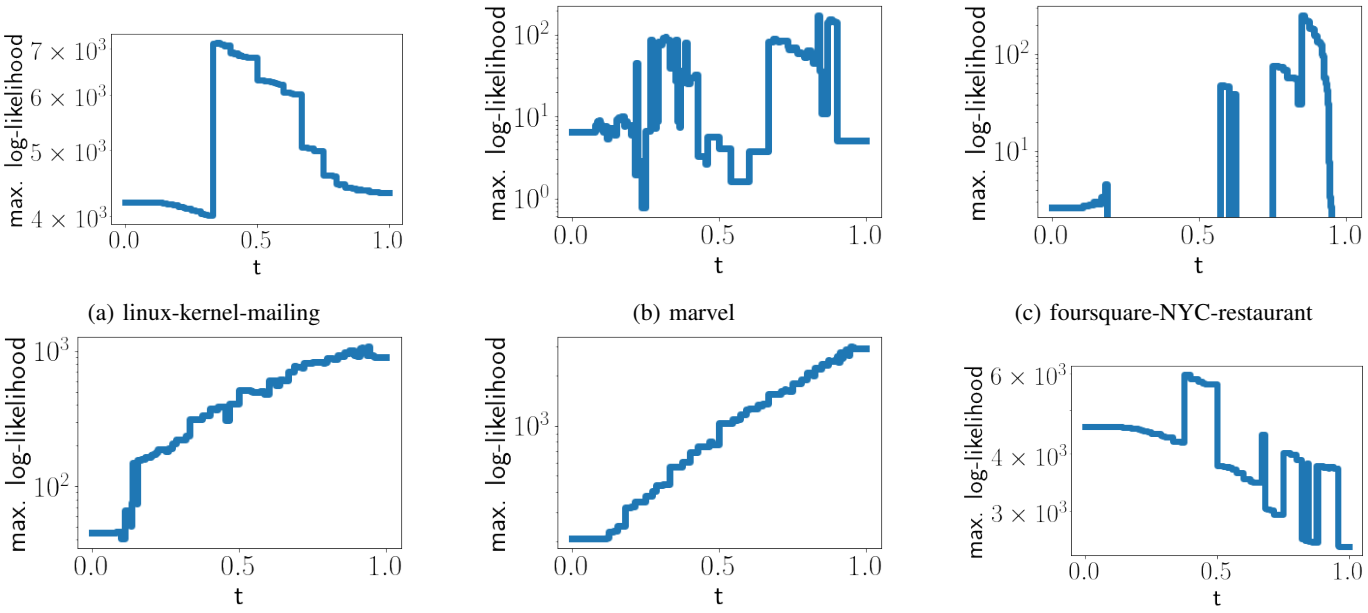


Fig. 22: (k, t)-hypercore sizes of the additional datasets.

TABLE IX: The full results of the HSMD distances including the additional datasets.

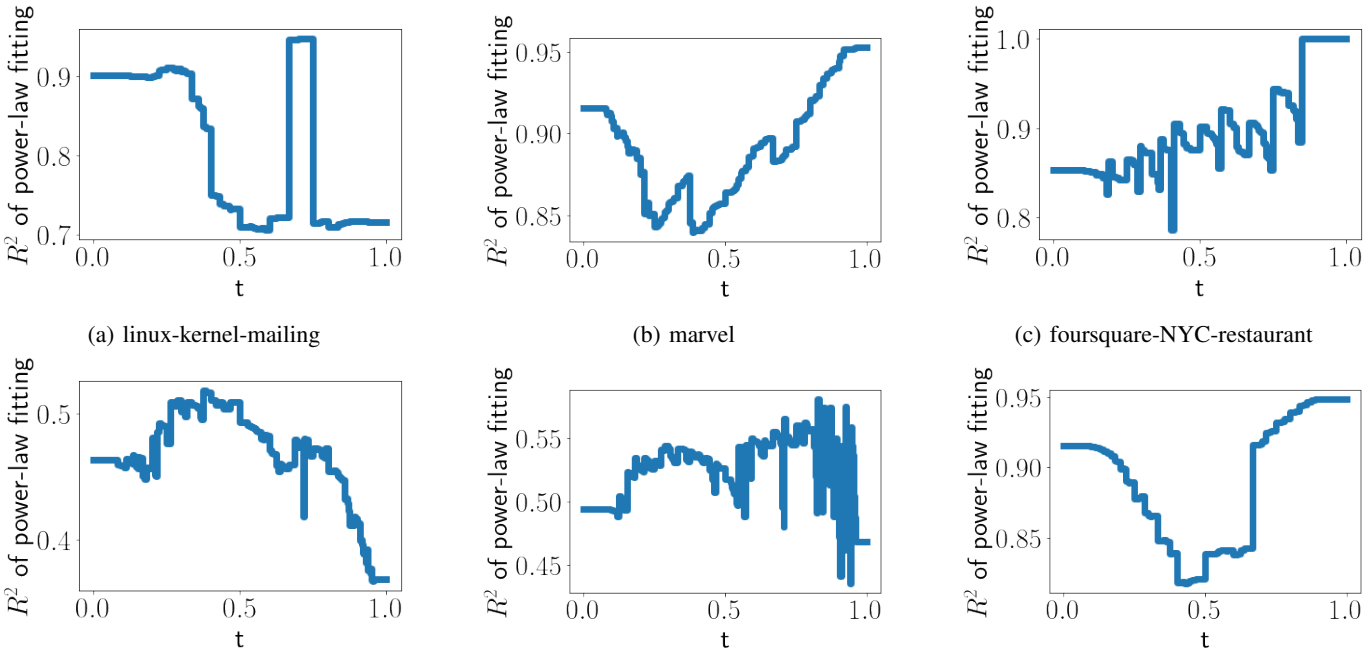
	coauth-DBLP	coauth-Geology	NDC-classes	NDC-substances	contact-high	contact-primary	email-Enron	email-Eu	tags-ubuntu	tags-math	tags-SO	threads-ubuntu	threads-math	threads-SO	linux-kernel-mailing	marvel	foursquare-NYC-restaurant	foursquare-NYC	foursquare-Tokyo	wikinews
coauth-DBLP	0.000	0.221	0.288	0.277	0.504	0.506	0.396	0.405	0.369	0.398	0.327	0.261	0.268	0.279	0.392	0.354	0.342	0.403	0.397	0.473
coauth-Geology	0.221	0.000	0.307	0.319	0.543	0.546	0.445	0.456	0.425	0.450	0.390	0.311	0.340	0.337	0.440	0.384	0.285	0.444	0.439	0.518
NDC-classes	0.288	0.307	0.000	0.211	0.488	0.487	0.379	0.390	0.350	0.377	0.313	0.341	0.305	0.351	0.370	0.292	0.401	0.380	0.378	0.456
NDC-substances	0.277	0.319	0.211	0.000	0.449	0.451	0.331	0.345	0.307	0.339	0.285	0.333	0.287	0.326	0.328	0.255	0.375	0.322	0.317	0.435
contact-high	0.504	0.543	0.488	0.449	0.000	0.103	0.300	0.292	0.338	0.322	0.386	0.442	0.410	0.425	0.313	0.482	0.578	0.322	0.339	0.291
contact-primary	0.506	0.546	0.487	0.451	0.103	0.000	0.294	0.282	0.323	0.301	0.372	0.437	0.411	0.431	0.300	0.488	0.595	0.331	0.352	0.254
email-Enron	0.396	0.445	0.379	0.331	0.300	0.294	0.000	0.135	0.192	0.173	0.254	0.333	0.288	0.319	0.139	0.360	0.496	0.156	0.189	0.321
email-Eu	0.405	0.456	0.390	0.345	0.292	0.282	0.135	0.000	0.190	0.167	0.253	0.332	0.292	0.324	0.158	0.373	0.512	0.193	0.224	0.293
tags-ubuntu	0.369	0.425	0.350	0.307	0.338	0.323	0.192	0.190	0.000	0.117	0.167	0.276	0.245	0.291	0.201	0.356	0.504	0.242	0.250	0.310
tags-math	0.398	0.450	0.377	0.339	0.322	0.301	0.173	0.167	0.117	0.000	0.203	0.307	0.281	0.319	0.180	0.376	0.529	0.236	0.259	0.294
tags-SO	0.327	0.390	0.313	0.285	0.386	0.372	0.254	0.253	0.167	0.203	0.000	0.242	0.207	0.274	0.253	0.357	0.486	0.300	0.300	0.340
threads-ubuntu	0.261	0.311	0.341	0.333	0.442	0.437	0.333	0.332	0.276	0.307	0.242	0.000	0.174	0.164	0.330	0.390	0.412	0.358	0.353	0.405
threads-math	0.268	0.340	0.305	0.287	0.410	0.411	0.288	0.292	0.245	0.281	0.207	0.174	0.000	0.169	0.296	0.355	0.409	0.303	0.292	0.399
threads-SO	0.279	0.337	0.351	0.326	0.425	0.431	0.319	0.324	0.291	0.319	0.274	0.164	0.169	0.000	0.326	0.383	0.386	0.326	0.313	0.421
linux-kernel-mailing	0.392	0.440	0.370	0.328	0.313	0.300	0.139	0.158	0.201	0.180	0.253	0.330	0.296	0.326	0.000	0.343	0.511	0.209	0.240	0.281
marvel	0.354	0.384	0.292	0.255	0.482	0.488	0.360	0.373	0.356	0.376	0.357	0.390	0.355	0.383	0.343	0.000	0.447	0.362	0.367	0.444
foursquare-NYC-restaurant	0.342	0.285	0.401	0.375	0.578	0.595	0.496	0.512	0.504	0.529	0.486	0.412	0.409	0.386	0.511	0.447	0.000	0.460	0.445	0.609
foursquare-NYC	0.403	0.444	0.380	0.322	0.322	0.331	0.156	0.193	0.242	0.236	0.300	0.358	0.303	0.326	0.209	0.362	0.460	0.000	0.109	0.396
foursquare-Tokyo	0.397	0.439	0.378	0.317	0.339	0.352	0.189	0.224	0.250	0.259	0.300	0.353	0.292	0.313	0.240	0.367	0.445	0.109	0.000	0.418
wikinews	0.473	0.518	0.456	0.435	0.291	0.254	0.321	0.293	0.310	0.294	0.340	0.405	0.399	0.421	0.281	0.444	0.609	0.396	0.418	0.000



(a) linux-kernel-mailing (b) marvel (c) foursquare-NYC-restaurant

(d) foursquare-NYC (e) foursquare-Tokyo (f) wikinews

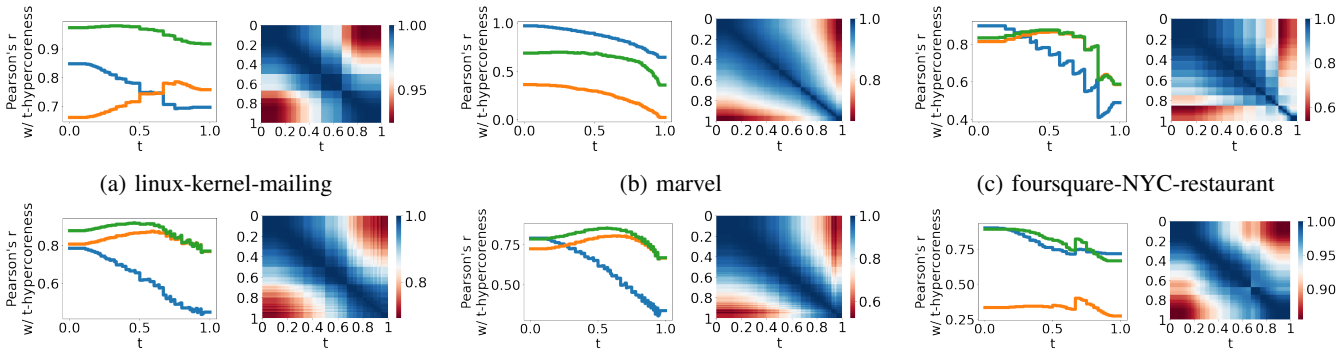
Fig. 23: **For each additional dataset and each t value, we report the log-likelihood ratio (R-value) of heavy-tailed distributions against the exponential distribution.**



(a) linux-kernel-mailing (b) marvel (c) foursquare-NYC-restaurant

(d) foursquare-NYC (e) foursquare-Tokyo (f) wikinews

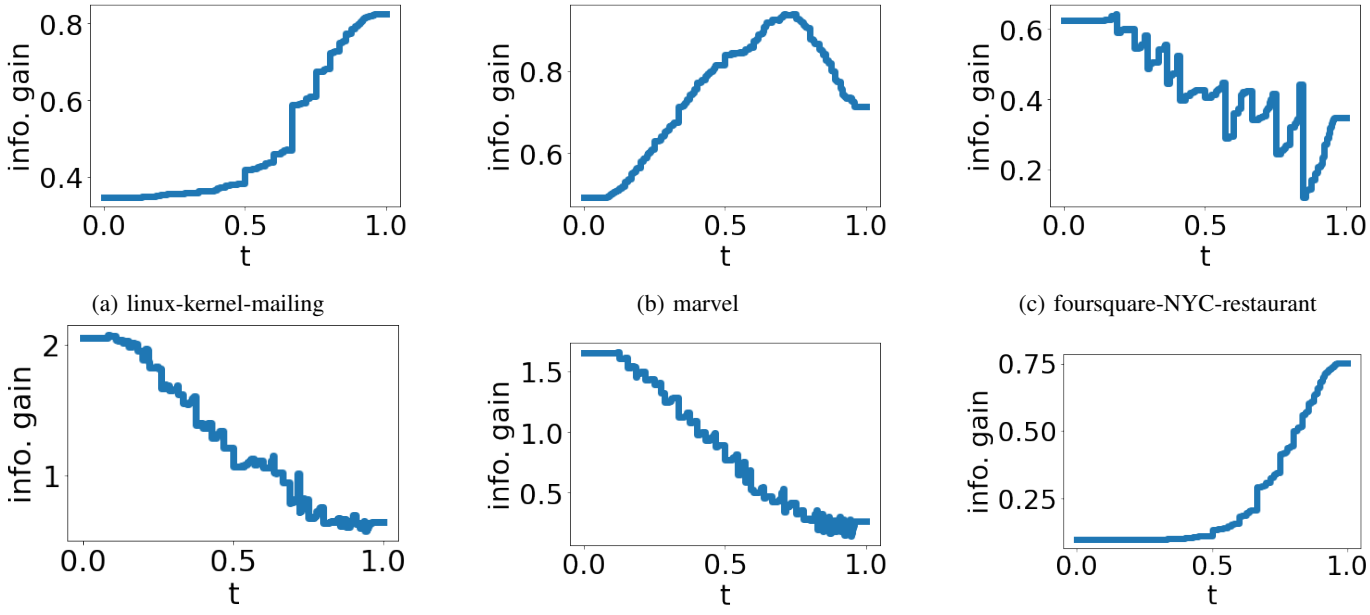
Fig. 24: **For each additional dataset and each t value, we show the R^2 value of the power-law fitting w.r.t the numbers of nodes with t-hypercoreness at least k with different k values.**



(a) linux-kernel-mailing (b) marvel (c) foursquare-NYC-restaurant

(d) foursquare-NYC (e) foursquare-Tokyo (f) wikinews

Fig. 25: **For the additional datasets, statistical difference also exists between t-hypercoreness and other centrality measures, as well as among t-hypercoreness with different t.** Left: the Pearson correlation coefficients between the t-hypercoreness sequences with different t and each of the degree and coreness sequences in the unweighted and weighted clique expansions. Right: the Pearson correlation coefficient between each pair of t-hypercoreness sequences.



(a) linux-kernel-mailing (b) marvel (c) foursquare-NYC-restaurant

(d) foursquare-NYC (e) foursquare-Tokyo (f) wikinews

Fig. 26: **For the additional datasets, t-hypercoreness also has substantial information gain over degree, and it provides distinct information depending on t.**

TABLE X: The full results of the Pearson's correlation coefficient w.r.t the information gain including the additional datasets.

	coauth-DBLP	coauth-Geology	NDC-classes	NDC-substances	contact-high	contact-primary	email-Enron	email-Eu	tags-ubuntu	tags-math	tags-SO	threads-ubuntu	threads-math	threads-SO	linux-kernel-mailing	marvel	foursquare-NYC-restaurant	foursquare-NYC	foursquare-Tokyo	wikinews
coauth-DBLP	1.000	0.482	0.709	0.681	-0.089	-0.674	0.270	0.098	0.257	0.254	0.269	0.424	0.240	0.292	0.285	0.894	-0.557	-0.631	-0.661	0.145
coauth-MAG-Geology	0.482	1.000	-0.257	-0.286	0.785	0.285	-0.561	-0.806	-0.698	-0.703	-0.696	-0.495	-0.708	-0.661	-0.698	0.229	0.316	0.320	0.296	-0.796
NDC-classes	0.709	-0.257	1.000	0.991	-0.736	-0.946	0.719	0.733	0.836	0.836	0.849	0.828	0.820	0.835	0.863	0.852	-0.914	-0.981	-0.989	0.781
NDC-substances	0.681	-0.286	0.991	1.000	-0.777	-0.953	0.688	0.748	0.868	0.868	0.878	0.863	0.852	0.868	0.880	0.831	-0.904	-0.969	-0.976	0.794
contact-high-school	-0.089	0.785	-0.736	-0.777	1.000	0.732	-0.666	-0.943	-0.973	-0.974	-0.970	-0.850	-0.970	-0.954	-0.956	-0.353	0.733	0.763	0.746	-0.953
contact-primary-school	-0.674	0.285	-0.946	-0.953	0.732	1.000	-0.770	-0.737	-0.835	-0.839	-0.851	-0.842	-0.821	-0.837	-0.859	-0.734	0.827	0.904	0.920	-0.782
email-Enron	0.270	-0.561	0.719	0.688	-0.666	-0.770	1.000	0.791	0.718	0.729	0.750	0.580	0.705	0.681	0.797	0.372	-0.653	-0.739	-0.745	0.822
email-Eu	0.098	-0.806	0.733	0.748	-0.943	-0.737	0.791	1.000	0.947	0.949	0.948	0.831	0.948	0.930	0.961	0.311	-0.689	-0.760	-0.747	0.982
tags-ask-ubuntu	0.257	-0.698	0.836	0.868	-0.973	-0.835	0.718	0.947	1.000	1.000	0.998	0.933	0.998	0.994	0.989	0.474	-0.782	-0.838	-0.830	0.965
tags-math-sx	0.254	-0.703	0.836	0.868	-0.974	-0.839	0.729	0.949	1.000	1.000	0.999	0.926	0.997	0.991	0.991	0.472	-0.784	-0.839	-0.833	0.968
tags-stack-overflow	0.269	-0.696	0.849	0.878	-0.970	-0.851	0.750	0.948	0.998	0.999	1.000	0.914	0.992	0.985	0.994	0.486	-0.796	-0.853	-0.847	0.971
threads-ask-ubuntu	0.424	-0.495	0.828	0.863	-0.850	-0.842	0.580	0.831	0.933	0.926	0.914	1.000	0.942	0.965	0.904	0.546	-0.715	-0.789	-0.788	0.841
threads-math-sx	0.240	-0.708	0.820	0.852	-0.970	-0.821	0.705	0.948	0.998	0.997	0.992	0.942	1.000	0.997	0.986	0.453	-0.768	-0.822	-0.812	0.963
threads-stack-overflow	0.292	-0.661	0.835	0.868	-0.954	-0.837	0.681	0.930	0.994	0.991	0.985	0.965	0.997	1.000	0.977	0.486	-0.769	-0.827	-0.820	0.944
linux-kernel-mailing	0.285	-0.698	0.863	0.880	-0.956	-0.859	0.797	0.961	0.989	0.991	0.994	0.904	0.986	0.977	1.000	0.494	-0.808	-0.870	-0.865	0.985
marvel	0.894	0.229	0.852	0.831	-0.353	-0.734	0.372	0.311	0.474	0.472	0.486	0.546	0.453	0.486	0.494	1.000	-0.777	-0.833	-0.843	0.364
foursquare-NYC-restaurant	-0.557	0.316	-0.914	-0.904	0.733	0.827	-0.653	-0.689	-0.782	-0.784	-0.796	-0.715	-0.768	-0.769	-0.808	-0.777	1.000	0.927	0.926	-0.743
foursquare-NYC	-0.631	0.320	-0.981	-0.969	0.763	0.904	-0.739	-0.760	-0.838	-0.839	-0.853	-0.789	-0.822	-0.827	-0.870	-0.833	0.927	1.000	0.990	-0.804
foursquare-Tokyo	-0.661	0.296	-0.989	-0.976	0.746	0.920	-0.745	-0.747	-0.830	-0.833	-0.847	-0.788	-0.812	-0.820	-0.865	-0.843	0.926	0.990	1.000	-0.796
wikinews	0.145	-0.796	0.781	0.794	-0.953	-0.782	0.822	0.982	0.965	0.968	0.971	0.841	0.963	0.944	0.985	0.364	-0.743	-0.804	-0.796	1.000

REFERENCES

[1] S. Limnios, G. Dasoulas, D. M. Thilikos, and M. Vazirgiannis, "Hcore-init: Neural network initialization based on graph degeneracy," in *ICPR*, 2021.
[2] Y. Zhang and S. Parthasarathy, "Extracting analyzing and visualizing triangle k-core motifs within networks," in *ICDE*, 2012.
[3] Y. Peng, Y. Zhang, W. Zhang, X. Lin, and L. Qin, "Efficient probabilistic k-core computation on uncertain graphs," in *ICDE*, 2018.
[4] K. Wang, X. Cao, X. Lin, W. Zhang, and L. Qin, "Efficient computing of radius-bounded k-cores," in *ICDE*, 2018.
[5] C. Zhang, F. Zhang, W. Zhang, B. Liu, Y. Zhang, L. Qin, and X. Lin, "Exploring finer granularity within the cores: Efficient (k, p)-core computation," in *ICDE*, 2020.
[6] F. Bonchi, A. Khan, and L. Severini, "Distance-generalized core decomposition," in *SIGMOD*, 2019.
[7] Q. Dai, R.-H. Li, L. Qin, G. Wang, W. Yang, Z. Zhang, and Y. Yuan, "Scaling up distance-generalized core decomposition," in *CIKM*, 2021.
[8] F. Zhang, Y. Zhang, L. Qin, W. Zhang, and X. Lin, "When engagement meets similarity: efficient (k, r)-core computation on social networks," in *PVLDB*, 2017.
[9] B. Liu, L. Yuan, X. Lin, L. Qin, W. Zhang, and J. Zhou, "Efficient (α, β)-core computation in bipartite graphs." in *VLDB Journal*, 2020.
[10] D. Ding, H. Li, Z. Huang, and N. Mamoulis, "Efficient fault-tolerant group recommendation using alpha-beta-core," in *CIKM*, 2017.
[11] K. Shin, T. Eliassi-Rad, and C. Faloutsos, "Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms," in *ICDM*, 2016.
[12] F. Victor, C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, "Alphacore: Data depth based core decomposition," in *KDD*, 2021.
[13] Z. Chen, L. Yuan, L. Han, and Z. Qian, "Higher-order truss decomposition in graphs," 2021.
[14] A. E. Sarıyüce and A. Pinar, "Peeling bipartite networks for dense subgraph discovery," in *WSDM*, 2018.
[15] K. Gabert, A. Pinar, and Ü. V. Çatalyürek, "A unifying framework to identify dense subgraphs on streams: Graph nuclei to hypergraph cores," in *WSDM*, 2021.
[16] G. Preti, G. De Francisci Morales, and F. Bonchi, "Strud: Truss decomposition of simplicial complexes," in *WWW*, 2021.
[17] G. Lee, M. Choe, and K. Shin, "How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators," in *WWW*, 2021.
[18] M. E. Aktas, T. Nguyen, S. Jawaid, R. Riza, and E. Akbas, "Identifying critical higher-order interactions in complex networks," in *Scientific Reports*, 2021.
[19] S.-e. Yoon, H. Song, K. Shin, and Y. Yi, "How much and when do we need higher-order information in hypergraphs? a case study on hyperedge prediction," in *WWW*, 2020.
[20] U. Chitra and B. Raphael, "Random walks on hypergraphs with edge-dependent vertex weights," in *ICML*, 2019.
[21] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, "Networks beyond pairwise interactions: structure and dynamics," in *Physics Reports*, 2020.
[22] L. Torres, A. S. Blevins, D. Bassett, and T. Eliassi-Rad, "The why, how, and when of representations for complex systems," 2021.
[23] H. C. Nguyen and H. Mamitsuka, "Learning on hypergraphs with sparsity," 2020.
[24] K. Hayashi, S. G. Aksoy, C. H. Park, and H. Park, "Hypergraph random walks, laplacians, and clustering," in *CIKM*, 2020.
[25] J. Huang, X. Liu, and Y. Song, "Hyper-path-based representation learning for hyper-networks," in *CIKM*, 2019.
[26] L. Bai, P. Ren, and E. R. Hancock, "A hypergraph kernel from isomorphism tests," in *ICPR*, 2014.
[27] H. Wu and M. K. Ng, "Hypergraph convolution on nodes-hyperedges network for semi-supervised node classification," in *TKDD*, 2022.
[28] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. N. Metaxas, "Unsupervised image categorization by hypergraph partition," in *TPAMI*, 2011.
[29] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, "Hypergraph learning with line expansion," in *arXiv*, 2020.
[30] Y. Dong, W. Sawin, and Y. Bengio, "Hnhn: Hypergraph networks with hyperedge neurons," in *arXiv*, 2020.
[31] X. Ouvrard, J.-M. L. Goff, and S. Marchand-Maillet, "Networks of collaborations: Hypergraph modeling and visualisation," in *arXiv*, 2017.
[32] J. Kunegis, "Konect: the koblenz network collection," in *WWW*, 2013.
[33] R. Alberich, J. Miro-Julia, and F. Rosselló, "Marvel universe looks almost like a real social network," *arXiv preprint cond-mat/0202174*, 2002.
[34] D. Yang, D. Zhang, Z. Yu, and Z. Yu, "Fine-grained preference-aware location search leveraging crowdsourced digital footprints from lbsns," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013.
[35] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2014.
[36] W. Foundation, "Wikimedia downloads," 2010. [Online]. Available: http://dumps.wikimedia.org/