Appendix

A. Proof of Property 1

Proof. We prove by contradiction. Suppose there are two distinct $(\mathcal{F}, \lambda, k)$-FoCores C and C'. According to Definition 2, for all $u \in C$ ($u \in C'$ resp.), we have $deg_C(u, l) \geq k$ ($deg_{C'}(u, l) \geq k$ resp.) on at least λ layers including $\mathcal{F}$. Therefore, for all $v \in C \cup C'$, we have $deg_{C \cup C'}(v, l) \geq \max(deg_C(v, l), deg_{C'}(v, l)) \geq k$ on at least λ layers including $\mathcal{F}$, so $C \cup C'$ is also a $(\mathcal{F}, \lambda, k)$-FoCore. It contradicts to the maximality of C and C'. Thus, the property holds. □

B. Proof of Property 2

Proof. Let v be an arbitrary vertex in $C^{k+1}_{\mathcal{F},\lambda}$. By Definition 2, v has at least λ support layers w.r.t. the minimum degree threshold $k+1$. Certainly, these layers support v w.r.t. the minimum degree threshold k. Moreover, in the induced subgraph $G[C^k_{\mathcal{F},\lambda}]$, the degree of v on each focus layer $l \in \mathcal{F}$ is at least $k+1$, which is certainly greater than k. Therefore, all vertices in $C^{k+1}_{\mathcal{F},\lambda}$ satisfy the constraints specified in Definition 2 w.r.t. the parameters $\mathcal{F}$, λ and k. Due to the maximality and the uniqueness of $C^k_{\mathcal{F},\lambda}$ (Property 1), we have $C^{k+1}_{\mathcal{F},\lambda} \subseteq C^k_{\mathcal{F},\lambda}$. □

C. Proof of Property 3

Proof. Let $C_1 = C^k_{\mathcal{F}_1,\lambda_1}$ and $C_2 = C^k_{\mathcal{F}_2,\lambda_2}$. By Definition 2, every vertex $v \in C_2$ has at least k neighbors on at least λ_2 layers (of course, on at least λ_1 layers), so C_2 satisfies the background constraint of the $(\mathcal{F}_1, \lambda_1, k)$-FoCore. By Definition 2, every vertex $v \in C_2$ satisfies $deg_{C_2}(v, l) \geq k$ for all $l \in \mathcal{F}_2$ (certainly, for all $l \in \mathcal{F}_1$), so C_2 also satisfies the focus constraint of the $(\mathcal{F}_1, \lambda_1, k)$-FoCore. Due to the maximality of C_1, we have $C_2 \subseteq C_1$. Thus, the property holds. □

D. Proof of Theorem 1

Proof. As will be presented in Section III, there exists algorithms that solve the FoCore decomposition problem in $f(|L|) \cdot |E|^{O(1)}$ time, where f is a function of $|L|$ that is independent of $|E|$. Thus, the theorem holds. □

E. Proof of Lemma 1

Proof. Let $k_1 = cn_{\mathcal{F}_1,\lambda_1}(v)$ and $k_2 = cn_{\mathcal{F}_2,\lambda_2}(v)$. We have $v \in C^{k_1}_{\mathcal{F}_1,\lambda_1}(v)$ and $v \in C^{k_2}_{\mathcal{F}_2,\lambda_2}(v)$. By Definition 4, k_1 is the maximum k such that $v \in C^k_{\mathcal{F}_1,\lambda_1}$. Since $(\mathcal{F}_1, \lambda_1) \sqsubseteq (\mathcal{F}_2, \lambda_2)$, we have $C^{k_2}_{\mathcal{F}_2,\lambda_2}(v) \subseteq C^{k_2}_{\mathcal{F}_1,\lambda_1}(v)$ by Property 3, so $v \in C^{k_2}_{\mathcal{F}_1,\lambda_1}(v)$. Thus, we have $k_1 \geq k_2$. □

F. Proof of Lemma 2

Proof. We prove the lemma by considering the above 3 cases:

Case 1: When $\mathcal{F} = \emptyset$, $\lambda = 1$ and $k = 1$, we set $R = V$, so $C^k_{\mathcal{F},\lambda} \subseteq V = R$.

Case 2: When $(\mathcal{F}, \lambda) \neq (\emptyset, 1)$ and $k = 1$, we have $R = \bigcap_{(\mathcal{F}',\lambda') \sqsubseteq (\mathcal{F},\lambda)} C^k_{\mathcal{F}',\lambda'}$. According to Property 3, we have $C^k_{\mathcal{F},\lambda} \subseteq C^k_{\mathcal{F}',\lambda'}$, so $C^k_{\mathcal{F},\lambda} \subseteq R$.

Case 3: When $k > 1$, we have $R = C^{k-1}_{\mathcal{F},\lambda}$. According to Property 2, we have $C^k_{\mathcal{F},\lambda} \subseteq R$. □

G. Proof of Theorem 2

Proof. (Necessity) If $cn_{\mathcal{F},\lambda}(v) = k$, we have $v \in C^k_{\mathcal{F},\lambda}$ and $|N(v) \cap C^k_{\mathcal{F},\lambda}| \geq k$. For all $w \in N(v) \cap C^k_{\mathcal{F},\lambda}$, we have $cn_{\mathcal{F},\lambda}(w) \geq k$, so the condition is met. Next, we prove by contradiction that k is the largest integer that satisfies the condition. Suppose $k' > k$ also satisfies the condition. Let $U \subseteq N(v)$ be the set of vertices that satisfy the condition w.r.t. k'. For $w \in U$, let C_w be the $(\mathcal{F}, \lambda, k')$-FoCore containing w. We have that $\bigcup_{w \in U} C_w \cup \{v\}$ forms a $(\mathcal{F}, \lambda, k')$-FoCore, so $cn_{\mathcal{F},\lambda}(v) = k'$, which contradicts with the fact that $cn_{\mathcal{F},\lambda}(v) = k$.

(Sufficiency) For each vertex $u \in N(v)$ that satisfies the condition, let C_u be the $(\mathcal{F}, \lambda, k)$-FoCore containing u. Consider $U = \bigcup C_u \cup \{v\}$. For all vertex $w \in U$, we have $deg_{G[U]}(w, l) \geq k$ for all $l \in \mathcal{F}$, and w has at least λ support layers in $G[U]$, so $v \in U \subseteq C^k_{\mathcal{F},\lambda}$. Since there is no $k' > k$ that satisfies the condition, we have $v \notin C^{k'}_{\mathcal{F},\lambda}$. Thus, $cn_{\mathcal{F},\lambda}(v) = k$. □

H. Proof of Lemma 3

Proof. By the definition of C^μ, C^μ is a $(\mathcal{F}_\mu, \lambda_\mu, k_\mu)$-FoCore, where $\mathcal{F}_\mu = \{l^\mu\}$, $\lambda_\mu = 1$, and k_μ is the maximum k such that the $(\mathcal{F}_\mu, \lambda_\mu, k)$-FoCore is not empty. Since C^* is the densest FoCore, we have $\rho(C^*) \geq \rho(C^\mu)$. □

I. Proof of Theorem 3

Proof. We have

$$\begin{aligned}
\rho(C^*) &\geq \rho(C^\mu) && \text{(by Lemma 3)} \\
&\geq \max_{l \in L} \frac{|E_l[C^\mu]|}{|C^\mu|} 1^\beta && \text{(by Eq. (2))} \\
&\geq \frac{1}{2} \max_{l \in L} \mu(C^\mu, l) && \text{(average degree $\geq$ min degree)} \\
&= \frac{1}{2} \mu(C^\mu, l^\mu) && \text{(by the definition of l^μ)} \\
&\geq \frac{1}{2} \mu(S^*_{SL}, l^*) && \text{(C^μ maximizes the min degree)} \\
&\geq \frac{1}{2} \cdot \frac{|E_{l^*}[S^*_{SL}]|}{|S^*_{SL}|} && \text{(by Lemma 4)} \\
&\geq \frac{1}{2|L|^\beta} \rho(S^*) && \text{(by Lemma 5)}
\end{aligned}$$

Thus, the theorem holds. □

J. Parallel Implementation of FoCore-VC

Vertex-centric algorithms are amenable to extension into parallel or distributed implementations. In this section, we introduce an experiment about the parallel implementation of FoCore-VC, which illustrates the decomposition time of the same combo vs. the number of threads employed.

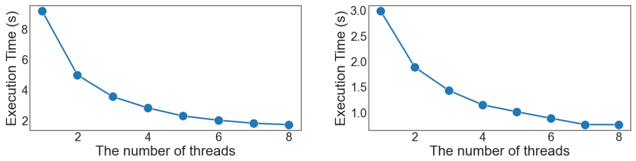
Algorithm 9 presents the parallel implementation, which operates within a Bulk Synchronous Parallel (BSP) paradigm. We can solve the FoCore decomposition problem in a parallel enviroment using FoCore-VC by replacing the `evalFoCoreNums-VC` in Line 7 into Algorithm 9. During each BSP round, Lines 4–10 execute updates on an active

Algorithm 9: `evalFoCoreNums-VC-parallel`

```
Input: A multilayer graph $G = (V, E, L)$, a combo $(\mathcal{F}, \lambda)$, and upper
       bounds $\overline{cn}_{\mathcal{F},\lambda}(v)$ of $cn_{\mathcal{F},\lambda}(v)$ for all $v \in V$
Output: An array $C$ where $C[v] = cn_{\mathcal{F},\lambda}(v)$ for all $v \in V$
1  $A \leftarrow V$; $B \leftarrow \emptyset$; $C \leftarrow \{\}$
2  for $v \in V$ do $C[v] \leftarrow \overline{cn}_{\mathcal{F},\lambda}(v)$
3  while $A \neq \emptyset$ do
4      for $v \in A$ in parallel do
5          $x \leftarrow$ estFoCoreNum$(G, \mathcal{F}, \lambda, C, v)$
6          if $x < C[v]$ then
7              $C[v] \leftarrow x$
8              for $u \in N(v)$ do
9                  if $C[u]$ does not satisfy the condition in Theorem 2 then
10                     $B \leftarrow B \cup \{u\}$
11     $A \leftarrow B$; $B \leftarrow \emptyset$
12 return $C$
```

vertex v and mark the active nodes for the subsequent round in the set B. Once all vertices in the active set A have been processed, Line 11 initiates synchronization and gathers the active nodes for the next round.

The proposed algorithms were implemented in C++ using OpenMP and compiled using clang++ (version 14.0.0). The experiment was conducted on an Apple M1 Pro CPU (3.2GHz, 8 cores) with 16GB of RAM. We employed Algorithm 9 to obtain the FoCore with combo $(\{\}, 1)$ while varying the number of threads. This particular combo, $(\{\}, 1)$, was chosen as it represents the slowest combo calculated by FoCore-VC among all possible combos.



(a) Amazon.

(b) DBLP.

Fig. 13. Execution time vs. number of threads for FoCore-VC.

We found that the speedup trends across different datasets exhibit similarity. Therefore, we provide examples using two datasets: Amazon and DBLP. As depicted in Fig. 13, as the number of threads increase, the execution time of Algorithm 9 will decrease. However, the computation cannot achieve perfect linear speedup due to the synchronization of the set of the active nodes in each BSP round.