APPENDIX A

A. Triangle-support lower bound for adaptively retained edges

Proof. Because $\hat{A}$ is symmetric and $(u,v) \in E$, $(\hat{A}^2)_{uv} = \hat{A}_{uu}\hat{A}_{uv} + \hat{A}_{uv}\hat{A}_{vv} + \sum_{w \in CN(u,v)} \hat{A}_{uw}\hat{A}_{wv}$. Using $\hat{A}_{uv} = 1/\sqrt{\hat{d}_u\hat{d}_v}$, $\hat{A}_{uu} = 1/\hat{d}_u$, $\hat{A}_{vv} = 1/\hat{d}_v$, and $\hat{A}_{uw}\hat{A}_{wv} = \frac{1}{\sqrt{\hat{d}_u\hat{d}_v}} \cdot \frac{1}{\hat{d}_w}$ for $w \in CN(u,v)$, we obtain

$$(\hat{A}^2 - \hat{A})_{uv} = \frac{1}{\sqrt{\hat{d}_u\hat{d}_v}} \left(\frac{1}{\hat{d}_u} + \frac{1}{\hat{d}_v} + \sum_{w \in CN(u,v)} \frac{1}{\hat{d}_w} - 1 \right).$$

The prefactor is positive, so the retention condition $(\hat{A}^2 - \hat{A})_{uv} > 0$ is equivalent to

$$\sum_{w \in CN(u,v)} \frac{1}{\hat{d}_w} \;>\; 1 - \frac{1}{\hat{d}_u} - \frac{1}{\hat{d}_v} \;=\; T(u,v).$$

Next, for any common neighbor $w \in CN(u,v)$, because we work with $A + I$, the node w is incident to at least the edges (w,u), (w,v), and the self-loop (w,w), hence $\hat{d}_w \geq 3$, which implies $\frac{1}{\hat{d}_w} \leq \frac{1}{3}$ for all $w \in CN(u,v)$. Summing over $CN(u,v)$ gives

$$\sum_{w \in CN(u,v)} \frac{1}{\hat{d}_w} \;\leq\; \frac{|CN(u,v)|}{3} \;=\; \frac{\mathrm{supp}(u,v)}{3}.$$

Combining above equations yields

$$\frac{\mathrm{supp}(u,v)}{3} \;>\; T(u,v) \quad \Rightarrow \quad \mathrm{supp}(u,v) \;>\; 3\,T(u,v).$$

Since $\mathrm{supp}(u,v)$ is an integer, we conclude

$$\mathrm{supp}(u,v) \;\geq\; \lfloor 3\,T(u,v) \rfloor + 1,$$

which match the claimed bound. □

B. Online Search Algorithm

Algorithm 1 operates as follows: Lines 1–2 normalize node embeddings and compute pairwise similarities on all edges in the original graph. Line 3 constructs the positive subgraph G^+ by retaining only those edges whose similarity exceeds the threshold τ. Lines 5–10 iteratively grow the community: In each iteration, Line 6 expands the community using BFS on G^+. If no more nodes can be reached and the community size is still smaller than the target $\mathcal{K}$ (Line 7), teleportation is performed (Line 8) by selecting the unvisited node with maximum similarity to the query node and adding it to the community. The process repeats until the community reaches the desired size, at which point the final set is returned.

This approach ensures that community expansion is guided primarily by structural connectivity and embedding similarity, while the teleportation step serves as a recovery strategy to overcome dead ends caused by sparsity or strong heterophily.

Algorithm 2 first estimates the global homophily ratio and selects a candidate set by semantic similarity to the query node. For each candidate, it computes a homophily-adaptive score that combines similarity and, if directly connected to the query, a topology-aware bonus or penalty. The $\mathcal{K}$ highest-scoring nodes (plus the query itself) form the returned community. This method is robust across varying homophily levels and avoids the pitfalls of methods that rely solely on topology or similarity.

Algorithm 1: Community Search via SCS (with teleport)

Input : Adjacency matrix A, embeddings $H \in \mathbb{R}^{n \times h}$, similarity threshold τ, query node $q \in V$, community size $\mathcal{K}$
Output: Community $\mathcal{C}_q \subseteq V$

1 Normalize embeddings: $\mathbf{Z} \leftarrow \mathrm{Norm}(H)$
2 Compute edge similarities $S[u,v] \leftarrow \mathbf{Z}_u \cdot \mathbf{Z}_v$ for all $(u,v) \in E$
3 Build positive graph G^+ by retaining $(u,v) \in E$ with $S[u,v] \geq \tau$
4 Initialize $\mathcal{C}_q = \{q\}$
5 **while** $|\mathcal{C}_q| < \mathcal{K}$ **do**
6 Expand $\mathcal{C}_q$ via BFS in G^+ until reach a dead-end
7 **if** $|\mathcal{C}_q| < \mathcal{K}$ **then**
8 Teleport to $u^* = \arg\max_{v \notin \mathcal{C}_q} S[q,v]$ to $\mathcal{C}_q$
9 **end if**
10 **end while**
11 **return** $\mathcal{C}_q$

Algorithm 2: Community Search via Adaptive Community Score (ACS)

Input : Query node q, embeddings $H \in \mathbb{R}^{n \times h}$, adjacency matrix A, community size $\mathcal{K}$, similarity-weighting parameter τ, hyperparameters: λ_{bonus}, λ_{penalty}, α
Output: Community $\mathcal{C}_q \subseteq V$

Estimate global homophily h_{edge} from random sample
$w_{\text{bon}} \leftarrow (1-\tau)h_{edge} \cdot \lambda_{\text{bonus}}$
$w_{\text{pen}} \leftarrow (1-\tau)(-(1-h_{edge}) \cdot \lambda_{\text{penalty}})$
Compute cosine similarities S_{qu} for all u
Select candidate set C as top-$(\alpha \cdot \mathcal{K})$ most similar nodes to q
for *each* $u \in C$ **do**
  if u *is adjacent to* q **then**
    score$[u] = S_{qu} + (w_{\text{bon}}$ if $h_{edge} \geq 0.5$ else $w_{\text{pen}})$
  end if
  else
    score$[u] = S_{qu}$
  end if
end for
Sort candidates in descending order of score
Return $\mathcal{C}_q =$ top-$\mathcal{K}$ nodes from sorted list (plus q)

C. Time Complexity Analysis

Let $n = |V|$, $m = |E|$, d be the input feature dimension, h the hidden size, k the number of hops, r the SVD rank, t the

TABLE I: Dataset statistics

Datasets	#nodes	#edges	#features	#classes	h_{edge}
Cora	2,708	5,429	1,433	7	0.8100
CiteSeer	3,327	4,732	3,703	6	0.7362
Photo	7,650	119,081	745	8	0.8272
Computers	13,752	245,861	767	10	0.7772
DBLP	17,716	52,867	1,639	4	0.8279
CS	18,333	81,894	6,805	15	0.8081
PubMed	19,717	44,338	500	3	0.8024
Reddit	232,956	116M	602	41	0.7817
Cornell	183	295	1,703	5	0.5669
Texas	183	309	1,703	5	0.4106
Wisconsin	251	499	1,703	5	0.4480
Chameleon	2,277	31,396	2,325	5	0.2299
Squirrel	5,201	198,423	2,089	5	0.2221
Film	7,600	33,544	931	5	0.3750
Roman	22,662	16,463	300	18	0.0469
Flickr	89,250	449,878	500	7	0.3195

number of training epochs, Q the number of queries, and $\mathcal{K}$ the target community size.

Offline stage. We compute a rank-r truncated SVD of the sparse normalized adjacency in $O(I\,m\,r)$ time using standard Krylov/Lanczos-type solvers, where I is the iteration count, and store $U_r, V_r \in \mathbb{R}^{n\times r}$ in $O(nr)$ memory. Projecting features costs $O(nrd)$, and constructing k hop-wise responses in the compressed subspace costs $O(k\,nrd)$. Model training costs $O\big(t\,(k\,m\,h + k\,n\,h^2)\big)$, where the first term is hop-wise propagation and the second term is the hop-wise linear transformations.

Online stage. For SCS, each query performs a graph traversal and scoring in $O(n + m)$ time (typically $O(m)$ on sparse graphs). For ACS, each query computes cosine similarities against all nodes in $O(nh)$ time, selects the top-M candidates ($M = \alpha\mathcal{K}$) via partial selection in $O(nh + M \log M)$ time, then re-scores and sorts the candidates in $O(M \log M)$ time (with adjacency checks in $O(M)$).

APPENDIX B

Table I summarizes the statistics of the datasets; we evaluate AdaptCS on 16 real-world graphs with varying levels of homophily. Eight datasets exhibit homophily, including Cora, CiteSeer, PubMed, Amazon-Computers, Amazon-Photo, Coauthor-CS, DBLP, and Reddit. The others show heterophily, including Cornell, Wisconsin, Texas, Film, Chameleon (corrected), Squirrel (corrected), Roman, and Flickr.