APPENDIX

A. Proof of Theorem 1

Proof. Consider a (γ_1, γ_2)-quasi-clique Q of graph G. For an arbitrary vertex v in Q, let us use V_j to denote the set of vertices whose shortest distance in G^u is j hops away from v, and assume that we can decompose Q into $V_0, V_1, \ldots, V_\ell$. Then, we have

$$
\begin{aligned}
|V_0| &= 1, && (4)\\
|V_1| &\geq \gamma_{max} \cdot (|Q| - 1), && (5)\\
|V_{i-1}| + |V_i| + |V_{i+1}| &\geq \gamma_{max} \cdot (|Q| - 1) + 1, && (6)\\
|V_{\ell-1}| + |V_\ell| &\geq \gamma_{max} \cdot (|Q| - 1) + 1, && (7)
\end{aligned}
$$

where Eq (4) is because $V_0 = \{v\}$; Eq (5) is because V_1 contain neighbors of v including γ_1 in-neighbors and γ_2 out-neighbors; Eq (6) is because for a vertex u in V_i, its neighbors must be within $V_{i-1} \cup V_i \cup V_{i+1}$ (recall that V_j's are defined over G^u), and u plus its neighbors contain at least $(\gamma_{max} \cdot (|Q| - 1) + 1)$ vertices; Eq (7) is because for a vertex u in V_ℓ, its neighbors must be within $V_{\ell-1} \cup V_\ell$. Then we can add the following formulas:

$$
\begin{aligned}
|V_0| + |V_1| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_0| + |V_1| + |V_2| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_1| + |V_2| + |V_3| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_2| + |V_3| + |V_4| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
&\cdots,\\
|V_{\ell-5}| + |V_{\ell-4}| + |V_{\ell-3}| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_{\ell-4}| + |V_{\ell-3}| + |V_{\ell-2}| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_{\ell-3}| + |V_{\ell-2}| + |V_{\ell-1}| &\geq \gamma_{max} \cdot (|Q| - 1) + 1,\\
|V_{\ell-1}| + |V_\ell| &\geq \gamma_{max} \cdot (|Q| - 1) + 1.
\end{aligned}
$$

After summation, we have $3 \cdot |Q| >$ the left hand side $\geq \ell \cdot (\gamma_{max} \cdot (|Q| - 1) + 1)$, so:

$$\ell < \frac{3|Q|}{\gamma_{max}(|Q| - 1) + 1},$$

which completes the proof since the vertex farthest from v in Q can be at most ℓ hops away. □

B. Proof of Theorem 2

Proof. Consider any two vertices u, v in a (γ_1, γ_2)-quasi-clique Q where $\gamma_1, \gamma_2 \geq 0.5$, we can easily show that u and v are at most 2 hops apart in G^u (c.f., Fig. 9). Specifically, we prove below that any two vertices u, v in Q cannot be more than 2 hops apart (i.e., cannot fall out of the 6 cases in Fig. 9).

Without loss of generality, we only consider the path from v to u where the first edge is outbound from v, i.e., Cases 1(a)–(c). Cases 2(a)–(c) are symmetric and can be similarly proved.

If v directly points to u, we are done since Case 1(a) occurs. Now assume that edge (v, u) does not exist in G, and we show that:

- **Case (I): edge** (u, v) **does not exist in** G, then both Case 1(b) and Case 1(c) should be satisfied. **(i) We first prove Case 1(b).** Note that $u \notin N^+(v)$ and $v \notin N^+(u)$. Since $\gamma_1 \geq 0.5$, u and v each points to at least $\lceil 0.5 \cdot (|Q| - 1) \rceil$ other vertices in Q, so they must share an out-neighbor; otherwise, there exist $2 \cdot \lceil 0.5 \cdot (|Q| - 1) \rceil \geq |Q| - 1$ vertices other than u and v, leading to a contradiction since there will be at least $(|Q| + 1)$ vertices in Q when adding u and v. **(ii) We next prove Case 1(c).** Note that $v \notin N^-(u)$ and $u \notin N^+(v)$. Since $\gamma_1 \geq 0.5$, v points to at least $\lceil 0.5 \cdot (|Q| - 1) \rceil$ other vertices in Q (here, u is excluded since $u \notin N^+(v)$); also since $\gamma_2 \geq 0.5$, u is pointed to by at least $\lceil 0.5 \cdot (|Q| - 1) \rceil$ other vertices in Q (here, v is excluded since $v \notin N^-(u)$). So, $N^+(v)$ and $N^-(u)$ must intersect as illustrated by Fig. 9 Case 1(c); otherwise, there will be $(|Q| + 1)$ vertices in Q when adding u and v.
- **Case (II): edge** (u, v) **exists in** G**,** then Case 1(c) should be satisfied. The proof is the same as (ii) above. Note that we cannot guarantee Case 1(b) anymore, since $v \in N^+(u)$, i.e., v can be one of the at least $\lceil 0.5 \cdot (|Q| - 1) \rceil$ neighbors of u, invalidating the prove for (i) above.

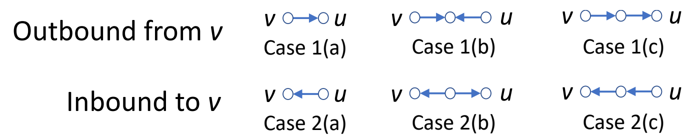


Fig. 9. Cases for Two-Hop Diameter Upper Bound

Symmetrically, consider the path from v to u where the first edge is inbound to v, i.e., Cases 2(a)–(c). If u directly points to v, we are done since Case 2(a) occurs. If edge (u, v) does not exist in G:

- **Case (III): edge** (v, u) **does not exist in** G, then both Case 2(b) and Case 2(c) should be satisfied. The proof is symmetric to Case (I) above and thus omitted.
- **Case (IV): edge** (v, u) **exists in** G, then Case 2(c) should be satisfied. The proof is symmetric to Case (II) above.

Putting the above discussions together, we obtain the following 4 cases, for each of which we explain how to exclude an impossible candidate u from $ext(S)$ given a vertex $v \in S$.

- **Case A:** $(v, u) \in E$ **and** $(u, v) \in E$**.** In this case, we always have $u \in ext(S)$.
- **Case B:** $(v, u) \notin E$ **and** $(u, v) \in E$**.** Based on Case (II) above, we have $u \in ext(S)$ only if a path $u \leftarrow w \leftarrow v$ exists in G for some $w \in V$ $(w \neq u, v)$.
- **Case C:** $(v, u) \in E$ **and** $(u, v) \notin E$**.** Based on Case (IV) above, we have $u \in ext(S)$ only if a path $u \rightarrow w \rightarrow v$ exists in G for some $w \in V$ $(w \neq u, v)$.
- **Case D:** $(v, u) \notin E$ **and** $(u, v) \notin E$**.** Based on Case (I) above, we have Condition (C1): $u \in ext(S)$ only if both Case 1(b) and Case 1(c) in Fig. 9 are satisfied. Similarly, based on Case (III) above, we have Condition (C2): $u \in ext(S)$ only if both Case 2(b) and Case 2(c) are satisfied. Combining both conditions, $u \in ext(S)$ only if there exist $w_1, w_2, w_3, w_4 \in V - \{u, v\}$ such that $u \leftarrow w_1 \leftarrow v$ and $u \leftarrow w_2 \rightarrow v$ and $u \rightarrow w_3 \leftarrow v$ and $u \rightarrow w_4 \rightarrow v$.

Once we have applied the above rules to prune $ext(S)$ to exclude invalid candidates u, let us abuse the notation to use G again to denote the resulting graph induced by $S \cup ext(S)$ after pruning. Note that we can apply this diameter-based pruning on the pruned G again, since some vertex w in Case B (resp. Case C) could have been pruned by Case C (resp. Case B) in the previous iteration, causing some required paths to disappear, further invalidating more vertices u from $ext(S)$. This pruning can be iteratively run over G.

Based on the above idea, Algorithm 1 computes the set of vertices in $ext(S)$ that are not 2-hop pruned by a vertex $v \in S$. Specifically, Line 1 computes O (resp. I) as the set of v's out-neighbors (resp. in-neighbors) u that belong to Case B (resp. Case C).

Then, Line 3 recovers S_O (resp. S_I) as the set of v's all non-pruned out-neighbors (resp. in-neighbors) w in Case B (resp. Case C) with path $v \to w \to u$ (resp. $v \leftarrow w \leftarrow u$). Note that $N^{\pm}(v) \subseteq ext(S)$ based on Case A so its vertices cannot be further pruned, so the iterative pruning is contributed by the shrink of sets O and I.

Next, Line 4 prunes away those vertices $u \in O$ (resp. $u \in I$) that cannot find a path $u \leftarrow w$ (resp. $u \to w$) for some non-pruned $w \in N^-(v)$ (resp. $w \in N^+(v)$), which is based on Case B (resp. Case C). Note that if O or I shrinks in Line 4, Line 5 will trigger another iteration of pruning. When the loop of Lines 2–5 exits, we have O (resp. I) being the remaining vertices $u \in ext(S)$ in Case B (resp. Case C) after iterative pruning.

Finally, Line 6 computes the set B of vertices where u satisfies Case D w.r.t. v, and Line 7 unions the 4 disjoint candidate sets that correspond to Cases A, B, C and D, respectively, to obtain the final 2-hop pruned $ext(S)$ for a vertex $v \in S$. We denote this set as $\mathbb{B}(v)$, which is returned by Line 7. □

C. Proof of Theorem 4

Proof. A valid (γ_1, γ_2)-quasi-clique $Q \subseteq V$ should contain at least τ_{size} vertices (i.e. $|Q| \geq \tau_{size}$), and therefore, for any $v \in Q$, its outdegree $d^+(v) \geq \lceil \gamma_1 \cdot (|Q|-1) \rceil \geq \lceil \gamma_1 \cdot (\tau_{size}-1) \rceil$ and indegree $d^-(v) \geq \lceil \gamma_2 \cdot (|Q|-1) \rceil \geq \lceil \gamma_2 \cdot (\tau_{size}-1) \rceil$. □

D. Degree-Based Pruning

Recall that $d^+_{V'}(v) = |N^+_{V'}(v)|$ and $d^-_{V'}(v) = |N^-_{V'}(v)|$. Thus, $d^+_S(v)$ (resp. $d^-_S(v)$) denotes the number of v's out-neighbors (resp. in-neighbors) in S, and $d^+_{ext(S)}(v)$ (resp. $d^-_{ext(S)}(v)$) denotes the number of v's out-neighbors (resp. in-neighbors) in $ext(S)$.

Theorem 8 (Type I Degree Pruning). *Given a vertex $u \in ext(S)$, if Condition (i): $d^+_S(u) + d^+_{ext(S)}(u) < \lceil \gamma_1 \cdot (|S| + d^+_{ext(S)}(u)) \rceil$ or Condition (ii): $d^-_S(u) + d^-_{ext(S)}(u) < \lceil \gamma_2 \cdot (|S| + d^-_{ext(S)}(u)) \rceil$ holds, then u can be pruned from $ext(S)$.*

This theorem is a result of the following lemma proven by [56].

Lemma 1. *If $a + n < \lceil \gamma \cdot (b + n) \rceil$ where $a, b, n \geq 0$, then $\forall i \in [0, n]$, we have $a + i < \lceil \gamma \cdot (b + i) \rceil$.*

Proof of Theorem 8. Theorem 8 follows since for any valid (γ_1, γ_2)-quasi-clique $Q = S \cup V'$ where $u \in V'$ and $V' \subseteq ext(S)$, we have

$$
\begin{aligned}
d^+_Q(u) &= d^+_S(u) + d^+_{V'}(u) && (8)\\
&< \lceil \gamma_1 \cdot (|S| + d^+_{V'}(u)) \rceil && (9)\\
&\leq \lceil \gamma_1 \cdot (|Q| - 1) \rceil, && (10)
\end{aligned}
$$

where Eq (8) is because $Q = S \cup V'$; Eq (9) is derived using Lemma 1, based on Condition (i) and the fact that $V' \subseteq ext(S)$; Eq (10) is because $\big(S \cup N^+_{V'}(u)\big) \subseteq \big(S \cup V' - \{u\}\big) = Q - \{u\}$. This result contradicts with the fact that Q is a (γ_1, γ_2)-quasi-clique. Condition (ii) is symmetric and a contradiction can be similarly derived. Therefore, if u satisfies either Condition (i) or (ii), we can safely prune u from $ext(S)$. □

Theorem 9 (Type II Degree Pruning). *Given vertex $v \in S$, if (1) $d^+_S(v) < \lceil \gamma_1 \cdot |S| \rceil$ and $d^+_{ext(S)}(v) = 0$, or (2) if $d^+_S(v) + d^+_{ext(S)}(v) < \lceil \gamma_1 \cdot (|S| - 1 + d^+_{ext(S)}(v)) \rceil$, then for any S' such that $S \subset S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.*

Given vertex $v \in S$, if (1) $d^-_S(v) < \lceil \gamma_2 \cdot |S| \rceil$ and $d^-_{ext(S)}(v) = 0$, or (2) if $d^-_S(v) + d^-_{ext(S)}(v) < \lceil \gamma_2 \cdot (|S| - 1 + d^-_{ext(S)}(v)) \rceil$, then for any S' such that $S \subset S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.

Proof. We hereby prove the pruning rule w.r.t. outdegrees, and the other rule w.r.t. indegrees is symmetric and can be similarly proved. First consider Condition (2), we have

$$
\begin{aligned}
d^+_Q(v) &= d^+_S(v) + d^+_{V'}(v) && (11)\\
&< \lceil \gamma_1 \cdot (|S| - 1 + d^+_{V'}(v)) \rceil && (12)\\
&\leq \lceil \gamma_1 \cdot (|Q| - 1) \rceil, && (13)
\end{aligned}
$$

where Eq (11) is because $Q = S \cup V'$; Eq (12) is derived using Lemma 1, based on Condition (2) and the fact that $V' \subseteq ext(S)$; Eq (13) is because $\big(S \cup N^+_{V'}(v)\big) \subseteq \big(S \cup V'\big) = Q$. This result contradicts with the fact that Q is a (γ_1, γ_2)-quasi-clique. Note that as long as we find one such $v \in S$, there is no need to extend S further. If $d^+_{ext(S)}(v) = 0$ in Condition (2), then we obtain $d^+_S(v) < \lceil \gamma_1 \cdot (|S| - 1) \rceil$ which is contained in Condition (1). Note that Condition (2) applies to the case $S = S'$ since i can be 0 in Lemma 1 (in contrast to Condition (1) to be explained below).

Now let us consider Condition (1). Condition (1) allows more effective pruning and is correct since for any valid quasi-clique $Q \supset S$ extended from S, we have $V' \neq \emptyset$ and

$$
\begin{aligned}
d^+_Q(v) &\leq d^+_S(v) + d^+_{ext(S)}(v) && (14)\\
&= d^+_S(v) && (15)\\
&< \lceil \gamma_1 \cdot (|Q| - 1) \rceil, && (16)
\end{aligned}
$$

where Eq (14) is because $Q = S \cup V'$ and $V' \subseteq ext(S)$; Eq (15) is because $d^+_{ext(S)}(v) = 0$ in Condition (1); Eq (16)

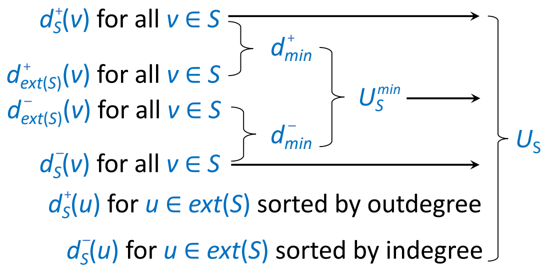


Fig. 10. Upper Bound Derivation

is because $d_S^+(v) < \lceil \gamma_1 \cdot |S| \rceil$ in Condition (1) and the fact that $|S| \leq |Q| - 1$ (recall that $V' \neq \emptyset$ and $Q = S \cup V'$). This result contradicts with the fact that Q is a (γ_1, γ_2)-quasi-clique. Note that the pruning of Condition (1) does not include the case where $S' = S$. □

E. Upper Bound Based Pruning

We next define an upper bound, denoted by U_S, on the number of vertices in $ext(S)$ that can be added to S concurrently to form a (γ_1, γ_2)-quasi-clique. The definition of U_S is based on $d_S^{\pm}(v)$ and $d_{ext(S)}^{\pm}(v)$ of all vertices $v \in S$ and on $d_S^{\pm}(u)$ of vertices $u \in ext(S)$ as summarized by Fig. 10, which we describe next.

We first define d_{min}^+ (resp. d_{min}^-) as the minimum outdegree (resp. minimum indegree) of any vertex in S, where the degrees are counted w.r.t. the other vertices in $S \cup ext(S)$ (c.f. Fig. 10):

$$d_{min}^+ = \min_{v \in S}\{d_S^+(v) + d_{ext(S)}^+(v)\}$$

$$d_{min}^- = \min_{v \in S}\{d_S^-(v) + d_{ext(S)}^-(v)\}$$

Now consider any quasi-clique S' such that $S \subseteq S' \subseteq (S \cup ext(S))$. For any $v \in S$, we have $d_S^+(v) + d_{ext(S)}^+(v) \geq d_{S'}^+(v) \geq \lceil \gamma_1 \cdot (|S'| - 1) \rceil$ and therefore, $d_{min}^+ \geq \lceil \gamma_1 \cdot (|S'| - 1) \rceil$. As a result, $\lfloor d_{min}^+/\gamma_1 \rfloor \geq \lfloor \lceil \gamma_1 \cdot (|S'| - 1) \rceil / \gamma_1 \rfloor \geq \lfloor \gamma_1 \cdot (|S'| - 1)/\gamma_1 \rfloor = |S'| - 1$, which gives the following upper bound on $|S'|$:

$$|S'| \leq \lfloor d_{min}^+/\gamma_1 \rfloor + 1. \tag{17}$$

We can similarly derive the other upper bound on $|S'|$ w.r.t. d_{min}^-:

$$|S'| \leq \lfloor d_{min}^-/\gamma_2 \rfloor + 1. \tag{18}$$

Combining Eq (17) and Eq (18), we obtain:

$$|S'| \leq \min\{\lfloor d_{min}^+/\gamma_1 \rfloor, \lfloor d_{min}^-/\gamma_2 \rfloor\} + 1. \tag{19}$$

Let us define U_S^{min} as an upper bound on the number of vertices from $ext(S)$ that can further extend S to form a valid quasi-clique. Using Eq (19) and the fact that vertices in S are already included in a quasi-clique to find (i.e., $S \subseteq S'$), we obtain (c.f. Fig. 10):

$$U_S^{min} = \min\{\lfloor d_{min}^+/\gamma_1 \rfloor, \lfloor d_{min}^-/\gamma_2 \rfloor\} + 1 - |S|. \tag{20}$$

We next tighten this upper bound using vertices in $ext(S) = \{u_1^+, u_2^+, \cdots u_n^+\}$[1], assuming that the vertices are listed in non-increasing order of outdegree $d_S^+(.)$. Similarly, we can also tighten this upper bound using vertices in $ext(S) = \{u_1^-, u_2^-, \cdots u_n^-\}$, assuming that the vertices are listed in non-increasing order of indegree $d_S^-(.)$. Then we have:

Lemma 2. *Given an integer k such that $1 \leq k \leq n$, if $\sum_{v \in S} d_S^+(v) + \sum_{i=1}^{k} d_S^-(u_i^-) < |S| \cdot \lceil \gamma_1(|S| + k - 1) \rceil$, then for any vertex set $Z \subseteq ext(S)$ with $|Z| = k$, $S \cup Z$ is not a (γ_1, γ_2)-quasi-clique.*

Proof. If S' is a (γ_1, γ_2)-quasi-clique, then for any $v \in S'$:

$$d_{S'}^+(v) \geq \lceil \gamma_1 \cdot (|S'| - 1) \rceil,$$

and therefore, for any $S \subseteq S'$, we have

$$\sum_{v \in S} d_{S'}^+(v) \geq |S| \cdot \lceil \gamma_1(|S'| - 1) \rceil. \tag{21}$$

Thus, to prove Lemma 2, we only need to show that

$$\sum_{v \in S} d_{S \cup Z}^+(v) < |S| \cdot \lceil \gamma_1(|S| + |Z| - 1) \rceil, \tag{22}$$

That is, Eq (21) is not satisfied for $S' = S \cup Z$, so a contradiction happens that invalidates S' from being a (γ_1, γ_2)-quasi-clique.

We now show that Eq (22) is correct below:

$$\begin{aligned}
\sum_{v \in S} d_{S \cup Z}^+(v) &= \sum_{v \in S} d_S^+(v) + \sum_{v \in S} d_Z^+(v) && (23)\\
&= \sum_{v \in S} d_S^+(v) + \sum_{u \in Z} d_S^-(u) && (24)\\
&\leq \sum_{v \in S} d_S^+(v) + \sum_{i=1}^{|Z|} d_S^-(u_i^-) && (25)\\
&< |S| \cdot \lceil \gamma_1(|S| + |Z| - 1) \rceil, && (26)
\end{aligned}$$

where Eq (23) is because $Z \subseteq ext(S)$ so $Z \cap S = \emptyset$; Eq (24) is because $\sum_{v \in S} d_Z^+(v) = \sum_{u \in Z} d_S^-(u) =$ the number of edges pointing from vertices in S to vertices in Z; Eq (25) is because $u_1^-, \cdots, u_{|Z|}^-$ are the k $(= |Z|)$ vertices with the highest $d_S^-(.)$ in $ext(S)$; Eq (26) is because of Lemma 2 $(k = |Z|)$. □

Symmetrically, we can also prove the following lemma:

Lemma 3. *Given an integer k such that $1 \leq k \leq n$, if $\sum_{v \in S} d_S^-(v) + \sum_{i=1}^{k} d_S^+(u_i^+) < |S| \cdot \lceil \gamma_2(|S| + k - 1) \rceil$, then for any vertex set $Z \subseteq ext(S)$ with $|Z| = k$, $S \cup Z$ is not a (γ_1, γ_2)-quasi-clique.*

Based on Lemma 2 and Lemma 3, we define a tightened upper bound U_S as follows (c.f. Fig. 10):

$$\begin{aligned}
U_S = \max\Bigg\{t \;\Bigg|\; &\left(1 \leq t \leq U_S^{min}\right) \bigwedge \left(\sum_{v \in S} d_S^+(v) + \sum_{i=1}^{t} d_S^-(u_i^-)\right.\\
&\left.\geq |S| \cdot \lceil \gamma_1(|S| + t - 1) \rceil\right) \bigwedge \left(\sum_{v \in S} d_S^-(v) + \sum_{i=1}^{t} d_S^+(u_i^+)\right.\\
&\left.\geq |S| \cdot \lceil \gamma_2(|S| + t - 1) \rceil\right)\Bigg\}. && (27)
\end{aligned}$$

[1]The superscript "+" is to indicate that vertices in $ext(S)$ are ordered by outdegree.

If such a t cannot be found, then S cannot be extended to generate a valid quasi-clique, which is a Type-II pruning. Otherwise, we further consider the 4 pruning rules to be described below which are based on U_S. Below, we only prove the theorems for outdegree-based upper bound pruning; the indegree-based rules are symmetric and can be similarly proved. We first describe Type-I pruning rules:

Theorem 10 (Type-I Outdegree Upper Bound Pruning). *Given a vertex $u \in ext(S)$, if $d_S^+(u)+U_S-1 < \lceil\gamma_1 \cdot (|S|+U_S-1)\rceil$, then u can be pruned from $ext(S)$.*

Proof. Consider any valid quasi-clique $Q = S \cup V'$ where $u \in V'$ and $V' \subseteq ext(S)$. If the condition in Theorem 10 holds, i.e., $d_S^+(u)+U_S-1 < \lceil\gamma_1 \cdot (|S|+U_S-1)\rceil$, then based on Lemma 1 and the fact that $|V'| \leq U_S$, we have:

$$d_S^+(u) + |V'| - 1 < \lceil\gamma_1 \cdot (|S| + |V'| - 1)\rceil = \lceil\gamma_1 \cdot (|Q| - 1)\rceil, \quad (28)$$

and therefore, $d_Q^+(u) = d_S^+(u) + d_{V'}^+(u) \leq d_S^+(u) + |V'| - 1 < \lceil\gamma_1 \cdot (|Q| - 1)\rceil$ (where the last step is due to Eq (28)), which contradicts with the fact that Q is a quasi-clique. □

Symmetrically, we can also prove the following theorem:

Theorem 11 (Type-I Indegree Upper Bound Pruning). *Given a vertex $u \in ext(S)$, if $d_S^-(u)+U_S-1 < \lceil\gamma_2 \cdot (|S|+U_S-1)\rceil$, then u can be pruned from $ext(S)$.*

We next describe Type-II pruning rules:

Theorem 12 (Type-II Outdegree Upper Bound Pruning). *Given a vertex $v \in S$, if $d_S^+(v) + U_S < \lceil\gamma_1 \cdot (|S| + U_S - 1)\rceil$, then for any S' such that $S \subseteq S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.*

Proof. Consider any valid quasi-clique $Q = S \cup V'$ where $v \in S$ and $V' \subseteq ext(S)$. If the condition in Theorem 12 holds, i.e., $d_S^+(v) + U_S < \lceil\gamma_1 \cdot (|S| + U_S - 1)\rceil$, then based on Lemma 1 and the fact that $|V'| \leq U_S$, we have:

$$d_S^+(v)+|V'| < \lceil\gamma_1 \cdot (|S|+|V'|-1)\rceil = \lceil\gamma_1 \cdot (|Q|-1)\rceil, \quad (29)$$

and therefore, $d_Q^+(v) = d_S^+(v) + d_{V'}^+(v) \leq d_S^+(v) + |V'| < \lceil\gamma_1 \cdot (|Q| - 1)\rceil$ (where the last step is due to Eq (29)), which contradicts with the fact that Q is a quasi-clique.

Since i can be 0 in Lemma 1, the pruning of Theorem 12 includes the case where $S' = S$, which is different from Theorem 9. □

Symmetrically, we can also prove the following theorem:

Theorem 13 (Type-II Indegree Upper Bound Pruning). *Given a vertex $v \in S$, if $d_S^-(v) + U_S < \lceil\gamma_2 \cdot (|S| + U_S - 1)\rceil$, then for any S' such that $S \subseteq S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.*

F. Lower Bound Based Pruning

Given a vertex set S, if some vertex $v \in S$ has $d_S^+(v) < \lceil\gamma_1 \cdot (|S| - 1)\rceil$ (or $d_S^-(v) < \lceil\gamma_2 \cdot (|S| - 1)\rceil$), then at least a certain number of vertices need to be added to S to increase the outdegree (or indegree) of v in order to form a (γ_1, γ_2)-quasi-clique. We denote this lower bound as L_{min}, which is

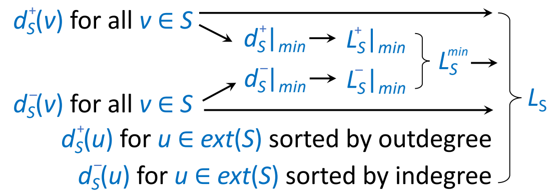


Fig. 11. Lower Bound Derivation

defined based on $d_S^{\pm}(v)$ of all vertices $v \in S$ and based on $d_S^{\pm}(u)$ of vertices $u \in ext(S)$ as summarized by Fig. 11, which we describe next.

We first define $d_S^+|_{min}$ as the minimum outdegree of any vertex in S and $d_S^-|_{min}$ as the minimum indegree of any vertex in S:

$$d_S^+|_{min} = \min_{v \in S} d_S^+(v), \qquad d_S^-|_{min} = \min_{v \in S} d_S^-(v)$$

Then, we can immediately derive the following two lower bounds:

$$L_S^+|_{min} = \min\{t \mid d_S^+|_{min} + t \geq \lceil\gamma_1 \cdot (|S| + t - 1)\rceil\} \quad (30)$$
$$L_S^-|_{min} = \min\{t \mid d_S^-|_{min} + t \geq \lceil\gamma_2 \cdot (|S| + t - 1)\rceil\} \quad (31)$$

Note that if even when all t newly added vertices are counted towards the degree of $v \in S$, the degree requirements w.r.t. γ_1 and γ_2 are still not satisfied, then we cannot make $S \cup Z$ (where $Z \subseteq ext(S)$ and $|Z| = t$) a valid quasi-clique, hence t is not valid. The lower bounds are taken as the smallest valid t.

To find such $L_S^+|_{min}$ (resp. $L_S^-|_{min}$), we check $t = 0, 1, \cdots, |ext(S)|$, and if none of them satisfies the inequality in Eq (30) (resp. Eq (31)), then S and its extensions cannot produce a valid quasi-clique, which is a Type-II pruning.

Otherwise, we obtain a lower bound:

$$L_S^{min} = \max\{L_S^+|_{min},\ L_S^-|_{min}\}. \quad (32)$$

We can further tighten this lower bound into L_S below using Lemma 2, assuming that vertices in $ext(S) = \{u_1^+, u_2^+, \cdots, u_n^+\}$ are listed in non-increasing order of $d_S^+(.)$, and $ext(S) = \{u_1^-, u_2^-, \cdots, u_n^-\}$ are listed in non-increasing order of $d_S^-(.)$:

$$L_S = \min\Bigg\{t \;\Bigg|\; \Big(L_S^{min} \leq t \leq n\Big) \bigwedge \Big(\sum_{v \in S} d_S^+(v) + \sum_{i=1}^{t} d_S^-(u_i^-) \geq |S| \cdot \lceil\gamma_1(|S| + t - 1)\rceil\Big) \bigwedge \Big(\sum_{v \in S} d_S^-(v) + \sum_{i=1}^{t} d_S^+(u_i^+) \geq |S| \cdot \lceil\gamma_2(|S| + t - 1)\rceil\Big)\Bigg\}. \quad (33)$$

If such a t cannot be found, then S cannot be extended to generate a valid quasi-clique, which is Type-II pruning. Otherwise, we further consider 4 pruning rules based on L_S which we list below. There, we only prove the theorems w.r.t. outdegree, since those w.r.t. indegree are symmetric. We first describe Type-I pruning rules:

Theorem 14 (Type-I Outdegree Lower Bound Pruning)**.** *Given a vertex $u \in ext(S)$, if $d_S^+(u)+d_{ext(S)}^+(u) < \lceil\gamma_1\cdot(|S|+L_S-1)\rceil$, then u can be pruned from $ext(S)$.*

Proof. Consider any valid quasi-clique $Q = S\cup V'$ where $u \in V'$ and $V' \subseteq ext(S)$. If the condition in Theorem 14 holds, we have $d_Q^+(u) = d_S^+(u) + d_{V'}^+(u) \leq d_S^+(u) + d_{ext(S)}^+(u) < \lceil\gamma_1 \cdot (|S| + L_S - 1)\rceil$ (due to the condition in Theorem 14) $\leq \lceil\gamma_1 \cdot (|Q| - 1)\rceil$ (since $L_S \leq |V'|$), which contradicts the fact that Q is a quasi-clique. □

Symmetrically, we can also prove the following theorem:

Theorem 15 (Type-I Indegree Lower Bound Pruning)**.** *Given a vertex $u \in ext(S)$, if $d_S^-(u)+d_{ext(S)}^-(u) < \lceil\gamma_2\cdot(|S|+L_S-1)\rceil$, then u can be pruned from $ext(S)$.*

We next describe Type-II pruning rules:

Theorem 16 (Type-II Outdegree Lower Bound Pruning)**.** *Given a vertex $v \in S$, if $d_S^+(v) + d_{ext(S)}^+(v) < \lceil\gamma_1 \cdot (|S| + L_S - 1)\rceil$, then for any S' such that $S \subseteq S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.*

Proof. Consider any valid quasi-clique $Q = S \cup V'$ where $v \in S$ and $V' \subseteq ext(S)$. If the condition in Theorem 16 holds, we have $d_Q^+(v) = d_S^+(v) + d_{V'}^+(v) \leq d_S^+(v) + d_{ext(S)}^+(v) < \lceil\gamma_1 \cdot (|S| + L_S - 1)\rceil$ (due to the condition in Theorem 16) $\leq \lceil\gamma_1 \cdot (|Q| - 1)\rceil$ (since $L_S \leq |V'|$), which contradicts the fact that Q is a quasi-clique. □

Symmetrically, we can also prove the following theorem:

Theorem 17 (Type-II Indegree Lower Bound Pruning)**.** *Given a vertex $v \in S$, if $d_S^-(v) + d_{ext(S)}^-(v) < \lceil\gamma_2 \cdot (|S| + L_S - 1)\rceil$, then for any S' such that $S \subseteq S' \subseteq (S \cup ext(S))$, $G(S')$ cannot be a (γ_1, γ_2)-quasi-clique.*

G. Proof of Theorems 5 and 6

Proof. This theorem is correct because if $u \in N_{ext(S)}^+(v)$ is not in S', then $d_{S'}^+(v) < d_S^+(v)+d_{ext(S)}^+(v) = \lceil\gamma_1\cdot(|S|+L_S-1)\rceil$ (due to Definition 3) $\leq \lceil\gamma_1 \cdot (|S'| - 1)\rceil$, which contradicts with the fact that S' is a (γ_1, γ_2)-quasi-clique. □

Symmetrically, we can also prove Theorems 6.

H. Proof of Theorem 7

We first prove that for any (γ_1, γ_2)-quasi-clique Q generated by extending S with vertices in $C_S^+(u)$, we have $d_{Q\cup u}^+(w) \geq \lceil\gamma_1 \cdot (|Q \cup u| - 1)\rceil = \lceil\gamma_1 \cdot |Q|\rceil$ for any vertex $w \in Q \cup u$. The other guarantee w.r.t. $C_S^-(u)$ is symmetric and can be similarly proved.

Proof. Recall from Fig. 3 that we only compute $C_S^+(u)$ for pruning if we have

$$d_S^+(u) \geq \lceil\gamma_1 \cdot |S|\rceil \tag{34}$$
$$d_S^+(v) \geq \lceil\gamma_1 \cdot |S|\rceil, \quad \forall\, v \in S \wedge v \notin N^-(u) \tag{35}$$

We divide the vertices $w \in Q \cup u$ in 3 disjoint sets (1) S, (2) $C_S^+(u) \subseteq ext(S)$, and (3) $\{u\}$ into 4 categories as follows, and prove that $d_{Q\cup u}^+(w) \geq \lceil\gamma_1 \cdot |Q|\rceil$ for any vertex w.

- **Case 1:** $w = u$ **(red in Fig. 3).** Then, we have

$$\begin{aligned} d_{Q\cup u}^+(u) &= d_S^+(u) + |Q| - |S| && (36)\\ &\geq \lceil\gamma_1 \cdot |S|\rceil + |Q| - |S| && (37)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil + |Q| - |Q| && (38)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil, \end{aligned}$$

  where Eq (36) is because u points to all the blue vertices in $C_S^+(u)$ (c.f. Fig. 3); Eq (37) is because of Eq (34); and Eq (38) is because $S \subseteq Q$ and $\lceil\gamma_1 - 1\rceil \leq 0$.
- **Case 2:** $w \in S$ **and** $w \notin N^-(u)$ **(green in Fig. 3).**

$$\begin{aligned} d_{Q\cup u}^+(w) &= d_S^+(w) + |Q| - |S| && (39)\\ &\geq \lceil\gamma_1 \cdot |S|\rceil + |Q| - |S| && (40)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil + |Q| - |Q| && (41)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil, \end{aligned}$$

  where Eq (39) is because all the green vertices point to all the blue vertices in $C_S^+(u)$ (c.f. Fig. 3); Eq (40) is because of Eq (35); and Eq (41) is because $S \subseteq Q$ and $\lceil\gamma_1 - 1\rceil \leq 0$.
- **Case 3:** $w \in S$ **and** $w \in N^-(u)$ **(yellow in Fig. 3).**

$$\begin{aligned} d_{Q\cup u}^+(w) &= d_Q^+(w) + 1 && (42)\\ &\geq \lceil\gamma_1 \cdot (|Q| - 1)\rceil + 1 && (43)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil, && (44) \end{aligned}$$

  where Eq (42) is because any yellow vertex in S should point to u (c.f. Fig. 3); Eq (43) is because Q is a (γ_1, γ_2)-quasi-clique; and Eq (44) is because $\lceil 1 - \gamma_1\rceil \geq 0$.
- **Case 4:** $w \in C_S^+(u)$ **(blue in Fig. 3).**

$$\begin{aligned} d_{Q\cup u}^+(w) &= d_Q^+(w) + 1 && (45)\\ &\geq \lceil\gamma_1 \cdot (|Q| - 1)\rceil + 1 && (46)\\ &\geq \lceil\gamma_1 \cdot |Q|\rceil, && (47) \end{aligned}$$

  where Eq (45) is because any blue vertex in $C_S^+(u)$ should point to u (c.f. Fig. 3); Eq (46) is because Q is a (γ_1, γ_2)-quasi-clique; and Eq (47) is because $\lceil 1 - \gamma_1\rceil \geq 0$.

As a special case, if all vertices in S points to u, then we do not have any vertex in Case 2, and our proof still holds. Here, we just need to compute $C_S^-(u) = N_{ext(S)}^+(u) \cap N_{ext(S)}^-(u)$ (c.f. Eq (1)). □

The other guarantee w.r.t. $C_S^-(u)$ (c.f. Eq (2)) is symmetric and can be similarly proved by reversing the directions of all edges. That is, for any (γ_1, γ_2)-quasi-clique Q generated by extending S with vertices in $C_S^-(u)$, $d_{Q\cup u}^-(w) \geq \lceil\gamma_2 \cdot |Q|\rceil$ for any $w \in Q \cup u$. Combining both guarantees, for any (γ_1, γ_2)-quasi-clique Q generated by extending S with vertices in $C_S(u) = C_S^+(u) \cap C_S^-(u)$, $Q \cup u$ is also a (γ_1, γ_2)-quasi-clique so Q is not maximal.

As for the degenerate special case when initially $S = \emptyset$, Eq (1) (resp. Eq (2)) becomes $C_S^+(u) = C_S^-(u) = N_{ext(S)}^+(u) \cap N_{ext(S)}^-(u)$ and all neighbors of u belong to $ext(S)$, so $C_S(u) = C_S^+(u) \cap C_S^-(u) = N_{ext(S)}^+(u) \cap$

Algorithm 4 *look_ahead*(S, $ext(S)$)

```
1: if |Q| < τ_size return FAIL
2: for each w ∈ S ∪ ext(S) do {here, we are iterating array A}
3:     if d_S^+(w) + d_ext(S)^+(w) < d_min^+(|S| + |ext(S)|) return FAIL
4:     if d_S^-(w) + d_ext(S)^-(w) < d_min^-(|S| + |ext(S)|) return FAIL
5:     if w ∈ ext(S) and d_ext(S)^𝔹(w) < |ext(S)| − 1 return FAIL
6: return SUCCEED
```

$N^-_{ext(S)}(u) = N^+(u) \cap N^-(u)$, i.e., we only need to find u as the vertex adjacent to the most number of bidirectional edges in G to maximize $|C_S(u)|$ for cover-vertex pruning. This is correct, since in our previous proof, there are no vertex in Cases 2 and 3 so no vertex breaks the requirement $d^+_{Q\cup u}(w) \geq \lceil \gamma_1 \cdot |Q| \rceil$ for any vertex $w \in Q \cup u$.

I. Degree Fields Maintained by Vertices in Array A

Each vertex object v in A (i.e., $S \cup ext(S)$) maintains five degrees (1) $d^+_S(v)$, (2) $d^-_S(v)$, (3) $d^+_{ext(S)}(v)$, (4) $d^-_{ext(S)}(v)$ and (5) the number of 2-hop neighbors $\mathbb{B}(v)$ that are in $ext(S)$, denoted by $d^{\mathbb{B}}_{ext(S)}(v) = |\mathbb{B}(v) \cap ext(S)|$. These five degree values are kept up-to-date whenever S and/or $ext(S)$ changes during the recursive mining, so that they can be accessed in $O(1)$ time when needed. Recall that these degree values are frequently needed when evaluating the conditions of our pruning rules, such as computing the bounds U_S and L_S as summarized in Fig. 10 and 11, so accessing them in $O(1)$ time is performance-critical. In fact, incrementally maintaining these degree values has a low cost: if a vertex v is moved or pruned, we only need to access those vertices in $N^+(v)$, $N^-(v)$, and $\mathbb{B}(v)$ to increment/decrement their degree values w.r.t. S and/or $ext(S)$.

J. Quasi-Clique Validation & Look-Ahead Pruning

Quasi-Clique Validation. Recall that τ_{size} is the size threshold for a valid quasi-clique. Let us define two functions:

$$d^+_{min}(size) = \lceil \gamma_1 \cdot (\max\{size, \tau_{size}\} - 1) \rceil, \quad (48)$$

$$d^-_{min}(size) = \lceil \gamma_2 \cdot (\max\{size, \tau_{size}\} - 1) \rceil. \quad (49)$$

Then, the following theorem directly follows from the definition of (γ_1, γ_2)-quasi-clique:

Theorem 18. *Let Q be a vertex set, then Q is a valid quasi-clique if and only if (i) $|Q| \geq \tau_{size}$ and (ii) for any vertex $v \in Q$, we have $d^+_Q(v) \geq d^+_{min}(|Q|)$ and $d^-_Q(v) \geq d^-_{min}(|Q|)$.*

The Look-Ahead Technique. This technique examines if $S \cup ext(S)$ gives a valid quasi-clique, and if so, we output it and avoid the unnecessary depth-first traversal of the subtree T_S. The rationale is that when $G(S \cup ext(S))$ is a valid quasi-clique and hence dense, traversing T_S can be expensive since pruning is less likely to be applicable during the traversal. In fact, when mining structures with a hereditary property such as k-plexes, look-ahead pruning is even essential since if $G(S \cup ext(S))$ is a k-plex, every node S in T_S is also a k-plex (and will thus be explored) but not maximal [57].

Algorithm 4 checks if $G(S \cup ext(S))$ is a (γ_1, γ_2)-quasi-clique, and returns *SUCCEED* if so to skip the traversal of subtree T_S. Specifically, we first make sure $|Q| \geq \tau_{size}$ in Line 1. Then, we check each vertex w in array $A = [S, ext(S)]$ one by one (Line 2), to examine the conditions of Theorem 18. If the conditions hold for all vertices in $S \cup ext(S)$, then $G(S \cup ext(S))$ is a valid quasi-clique so we return *SUCCEED* in Line 6; while if they do not hold for some vertex w, we return *FAIL* immediately as in Lines 3 and 4.

Line 5 provides an additional pruning if w is a vertex in $ext(S)$: if $d^{\mathbb{B}}_{ext(S)}(w) < |ext(S)| - 1$ (recall that we keep $d^{\mathbb{B}}_{ext(S)}(w) = |\mathbb{B}(w) \cap ext(S)|$ with w), then there must exist another vertex $u \in ext(S)$ such that $u \notin \mathbb{B}(w)$, so u and w cannot appear together in any valid quasi-clique, including $G(S \cup ext(S))$. Note that we do not need to consider $w \in S$ since when we add w into S, we always make sure that $w \in \mathbb{B}(v)$ for any $v \in S$, and we always prune away those vertices in $ext(S)$ that are not in $\mathbb{B}(w)$.

Algorithm 5 *iterative_bound_pruning*(S, $ext(S)$)

```
1: compute bounds U_S and L_S
2: conduct degree-based, upper- and lower-bound based pruning
   using Type-II pruning rules for every v ∈ S
3: if S is Type-II pruned do    return PRUNED
4: if L_S ≤ U_S then
5:     conduct degree-based, upper- and lower-bound based pruning
       using Type-I pruning rules for every u ∈ ext(S)
6:     while ext(S) ≠ ∅ and ext(S) shrank do
7:         update d_ext(S)^+(.) and d_ext(S)^-(.)
8:         compute bounds U_S and L_S
9:         conduct degree-based, upper- and lower-bound based prun-
           ing using Type-II pruning rules for every v ∈ S
10:        if S is Type-II pruned then    return PRUNED
11:        if L_S > U_S then    ext(S) ← ∅;  return NOT_PRUNED
12:        conduct degree-based, upper- and lower-bound based prun-
           ing using Type-I pruning rules for every u ∈ ext(S)
13: else    ext(S) ← ∅
14: return NOT_PRUNED
```

K. Iterative Bound-Based Pruning

Whenever we remove a candidate vertex from $ext(S)$ and/or add a candidate vertex to S, the degrees of the vertices in array A w.r.t. S and $ext(S)$ would be incrementally updated, creating new opportunities for degree-based pruning (c.f. Appendix D). Moreover, the degree updates would also cause the bounds L_S and U_S to be updated (c.f. Fig. 10 and 11), creating new opportunities for upper bound pruning (c.f. Appendix E) and lower bound based pruning (c.f. Appendix F).

Note that some of the above pruning rules could be Type I rules, causing $ext(S)$ to shrink, which in turn reduces $d^+_{ext(S)}(.)$ and $d^-_{ext(S)}(.)$ and thus triggers another round of bound-based pruning.

Algorithm 5 shows the process of iterative bound-based pruning. Specifically, Line 1 first computes U_S and L_S following the procedures summarized in Fig. 10 and 11. Type-II pruning may occur during the process of computing U_S and L_S (c.f., the paragraphs after Eq (27), after Eq (33) and before Eq (32)). If Type-II pruning occurs in Line 1, Algorithm 5 will return tag *PRUNED* directly so that the main algorithm will skip subtree T_S. Otherwise, Line 2 conducts degree-based Type-II pruning (i.e., Theorem 9), upper-bound based Type-II pruning (i.e., Theorems 12 and 13), and lower-bound

TABLE VIII
EFFECT OF QUASI-CLIQUE PARAMETERS ON *Bitcoin*

τ_{size}	γ_1	γ_2	Runtime (all rules)	Runtime (w/o cover)	# Maximal
	0.67		72.32	16.29	166,014
	0.68		68.53	15.49	166,014
	0.69		56.13	14.40	174,785
10	0.7	0.6	36.71	9.88	287,139
	0.71		21.40	3.28	24,962
	0.72		16.20	2.78	34,470
	0.73		12.09	1.88	9,446
		0.57	54.32	13.39	261,451
		0.58	43.23	12.09	281,868
		0.59	38.20	8.99	287,139
10	0.7	0.6	36.71	9.88	287,139
		0.61	23.40	4.18	72,215
		0.62	22.71	4.39	72,333
		0.63	23.42	4.29	72,333
7			41.65	12.54	320,836
8			41.82	12.82	320,763
9	0.7	0.6	39.44	11.21	289,114
10			36.71	9.88	287,139
11			36.00	9.00	287,138
12			23.83	1.99	24,344

TABLE IX
EFFECT OF QUASI-CLIQUE PARAMETERS ON *Epinions*

τ_{size}	γ_1	γ_2	Runtime (all rules)	Runtime (w/o cover)	# Maximal
	0.77		15.54	6.24	469
	0.78		15.86	6.15	469
	0.79		15.95	5.94	469
20	0.8	0.9	15.54	6.82	469
	0.81		10.84	5.05	345
	0.82		10.84	4.25	345
	0.83		10.63	4.74	345
		0.87	46.49	11.27	2,669
		0.88	34.88	9.28	2,669
		0.89	18.48	7.37	2,669
20	0.8	0.9	15.54	6.82	469
		0.91	6.24	4.24	0
		0.92	5.44	3.13	0
		0.93	3.84	2.74	0
17			18.60	7.14	687
18			17.11	7.62	477
19	0.8	0.9	17.28	6.38	473
20			15.54	6.82	469
21			14.95	5.84	469
22			10.48	4.07	24

based Type-II pruning (i.e., Theorems 16 and 17). Algorithm 5 returns *PRUNED* directly in Line 3 if S is Type-II pruned. Otherwise, if we find $L_S > U_S$ in Line 4 meaning that S cannot be expanded further into a valid quasi-clique, we set $ext(S) \leftarrow \emptyset$ in Line 13 and return *NOT_PRUNED* in Line 14 to indicate that T_S is not Type-II pruned. Note that if *iterative_bound_pruning*$(S, ext(S))$ returns *NOT_PRUNED* but $ext(S)$ has been set to $\emptyset$, we still need to examine if $G(S)$ is a valid quasi-clique but not any other descendant in T_S.

Otherwise, Line 5 then conducts degree-based Type-I pruning (i.e., Theorem 8), upper-bound based Type-I pruning (i.e., Theorems 10 and 11), and lower-bound based Type-I pruning (i.e., Theorems 14 and 15). If some vertices have been Type-I pruned from $ext(S)$, and $ext(S) \neq \emptyset$ (Line 6), then since the degrees $d^+_{ext(S)}(.)$ and $d^-_{ext(S)}(.)$ may decrease triggering the update of U_S and L_S and hence more pruning opportunities, we enter the iterative pruning procedure given by Lines 6–12. Specifically, Line 7 updates $d^+_{ext(S)}(.)$ and $d^-_{ext(S)}(.)$, and Line 8 updates U_S and L_S, to reflect the removal of Type-I pruned vertices from $ext(S)$. Then, Line 9 conducts Type-II pruning once more, followed by Line 12 for Type-I pruning once more, and the iterative pruning repeats by going back to Line 6 for another iteration of pruning.

Note that Line 10 and Line 11 help skip unnecessarily executing the expensive checking in Line 12 before checking the loop-exiting conditions in Line 6. Another detail not shown in Algorithm 5 is with Condition (1) in Theorem 9 which Type-II prunes T_S except for S itself, in which case instead of returning *PRUNED*, we set $ext(S) = \emptyset$ and return *NOT_PRUNED* so $G(S)$ will still be examined.

L. *Effect of Quasi-Clique Parameters*

Effect of Quasi-Clique Parameters. Recall that we tuned the quasi-clique parameters $(\tau_{size}, \gamma_1, \gamma_2)$ and used them in our experiments by default. Here, we show how the mining time and number of results vary as the parameters change, using *Bitcoin* and *Epinions*.

We show the effect of changing the quasi-clique parameters $(\tau_{size}, \gamma_1, \gamma_2)$ by varying one parameter while fixing the other two. Table VIII shows the results on *Bitcoin* for illustration. We can see that a small change of a parameter value can change the number of results a lot. For example, when changing γ_1 from $(10, 0.7)$ to $(10, 0.71)$, the result number decreases from 287,139 to 24,962 due to the stricter density requirements. The change of result number is, however, not monotonic. For example, when changing γ_1 from $(10, 0.69)$ to $(10, 0.70)$, the result number actually increases which might appear counter-intuitive. The reason is that some previously valid quasi-cliques get split into multiple smaller quasi-cliques rather than being eliminated. Table IX shows the results

TABLE X
ABLATION STUDY: ALL BUT ONE

Algorithm	PolBlogs		Epinions		Google	
	Runtime	Memory	Runtime	Memory	Runtime	Memory
full version	8.68	79	15.66	479	0.77	564
w/o lookahead	7.96	72	14.74	520	0.78	611
w/o critical	9.08	88	16.25	532	0.78	583
w/o cover	1.46	31	6.82	290	0.79	565
w/o bound	48.90	184	268.40	924	0.76	572

Algorithm	Baidu		USA Road		ClueWeb	
	Runtime	Memory	Runtime	Memory	Runtime	Memory
full version	9.21	1,524	9.81	15,170	172.85	25,5371
w/o lookahead	10.58	1,436	10.94	15,250	175.42	25,5690
w/o critical	10.20	1,509	10.79	15,223	171.75	25,5349
w/o cover	8.82	1,576	11.06	14,990	174.79	25,5437
w/o bound	10.67	1,439	10.52	15,041	176.47	25,5513

TABLE XI
ABLATION STUDY: INCREMENTAL ADDITION

Algorithm	MathOverflow		PolBlogs		Epinions	
	Runtime	Memory	Runtime	Memory	Runtime	Memory
baseline	385.29	798	2.30	45	94.76	665
+bound	22.82	217	1.40	24	5.81	291
+critical	17.51	186	1.40	23	6.71	273
+lookahead	17.05	209	1.46	31	6.82	290
+cover	551.49	738	8.68	79	15.66	479

Algorithm	Baidu		USA Road		ClueWeb	
	Runtime	Memory	Runtime	Memory	Runtime	Memory
baseline	20.57	1,549	10.34	15,575	191.63	257,502
+bound	19.60	1,473	9.62	15,450	199.79	255,568
+critical	19.97	1,604	8.71	15,433	196.47	255,567
+lookahead	8.82	1,576	9.82	15,225	174.79	255,437
+cover	9.21	1,524	9.81	15,170	172.85	255,371

on *Epinions* for illustration, and we can obtain a similar observation.

M. Ablation Study

We report the ablation study results of those algorithm variants which use all but one technique on the other 6 datasets in Table X. We can see that bound-based pruning is very effective, without which the running time can be much longer as on *PolBlogs* and *Epinions*. Also, our recommended configuration "w/o cover" is consistently the fastest or near-fastest, and exhibits much better performance on *PolBlogs* and *Epinions* than other configurations.

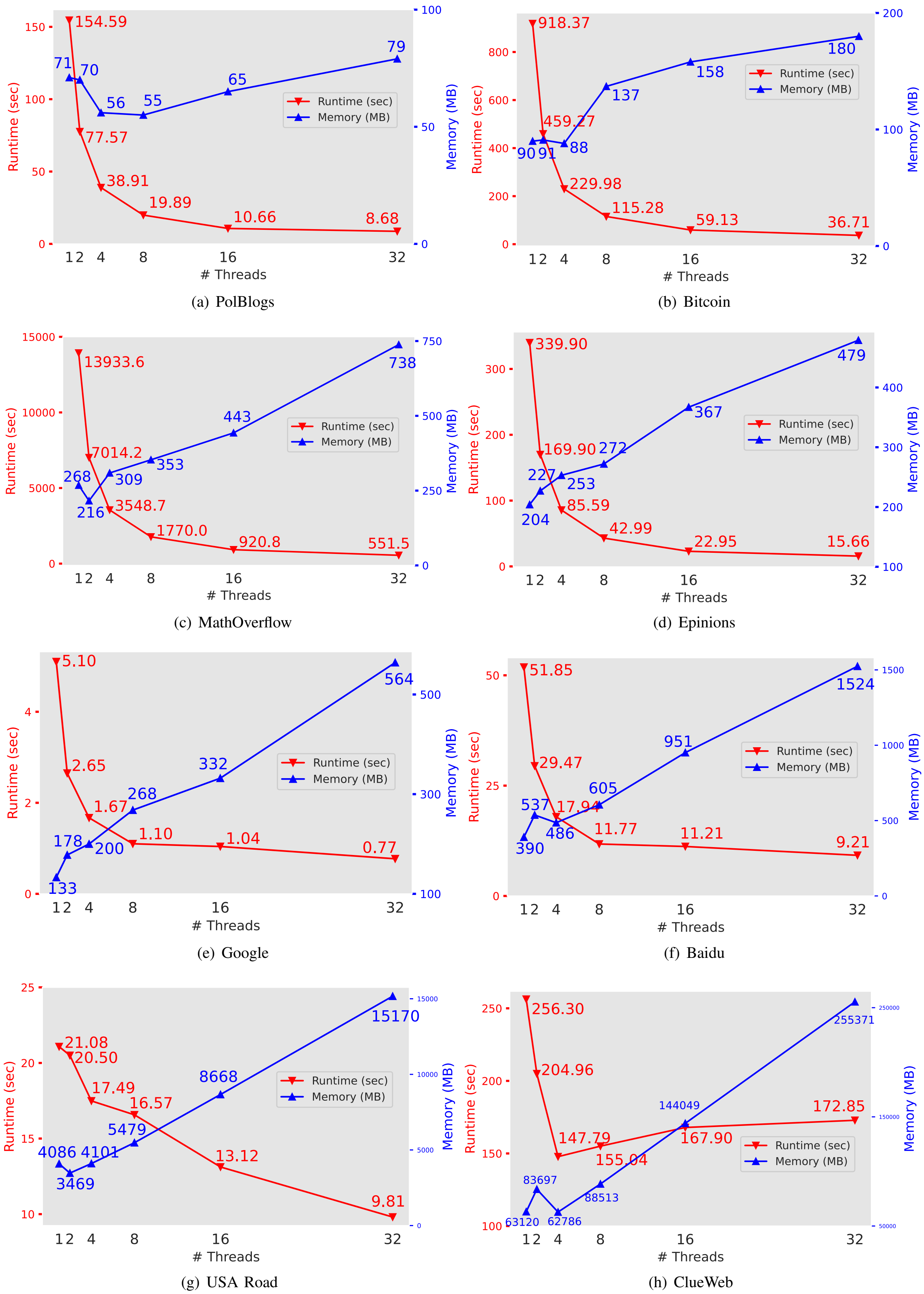
We also report our algorithm variants starting from a baseline with basic diameter-based, size-threshold, and degree-based pruning, and incrementally adding bound-based, critical-vertex, look-ahead, and cover-vertex pruning, one at a time. This gives algorithm variants denoted by "baseline," "+bound," "+critical," "+lookahead," and "+cover."

Table XI reports the results on the other 6 datasets. We can see that bound-based pruning significantly speeds up the baseline, especially on *MathOverflow* and *Epinions*. As we have discussed previously, adding look-ahead and cover-vertex pruning generally slows down the computation but can speed up web graphs such as *Baidu* and *ClueWeb* (as well as *Google* as shown in Table VI).

N. Scalability

We report the scalability study results of our parallel algorithm with all pruning rules enabled in Fig. 12, and we report the scalability study results of our parallel algorithm with all but cover-vertex pruning in Fig. 13.

We can observe on most datasets that the running time almost halves each time the number of threads doubles, except that the time curve hits a floor higher than 0 on *Baidu* and *ClueWeb* as the memory-bound computing of $\mathbb{B}(v)$ dominates the runtime.



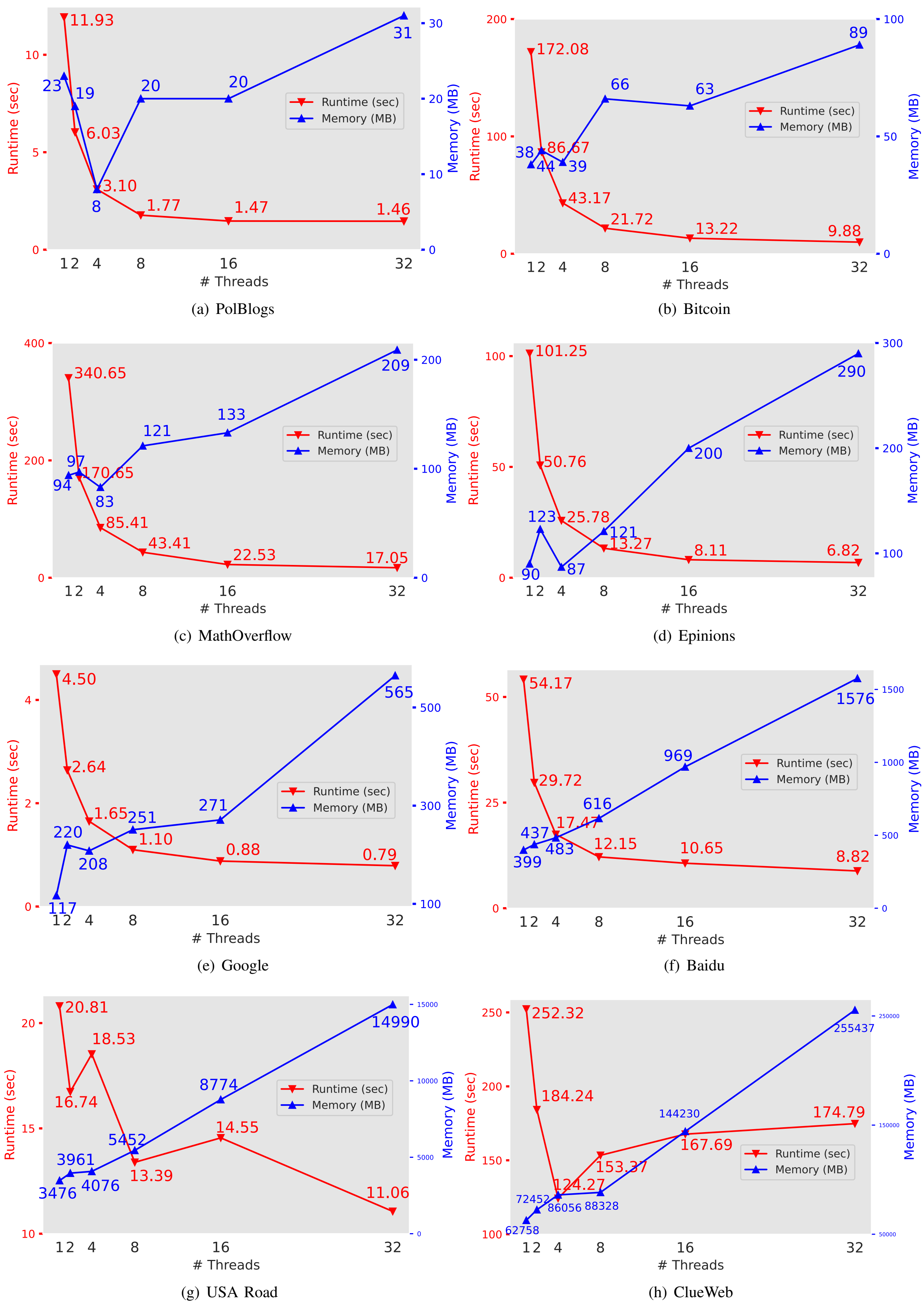
(a) PolBlogs (b) Bitcoin

(c) MathOverflow (d) Epinions

(e) Google (f) Baidu

(g) USA Road (h) ClueWeb

Fig. 12. Scalability of "full version"



(a) PolBlogs

(b) Bitcoin

(c) MathOverflow

(d) Epinions

(e) Google

(f) Baidu

(g) USA Road

(h) ClueWeb

Fig. 13. Scalability of “w/o cover”