APPENDIX A
ADDITIONAL PROOFS OF THEOREM

A. Theorem of BQC Diameter

In BQC definition, γ_1 and γ_2 control the cohesiveness of BQC: with larger γ_1 or γ_2, each node connects to more other nodes and thus the BQC is more cohesive. In graph analysis, one common measure of evaluating subgraph graph compactness is graph diameter [11], [13]. The following theorem discloses the upper bound of BQC diameter for arbitrary (γ_1, γ_2).

Theorem 9. *(Diameter of BQC). Let $Q = \{Q_L, Q_R\}$ be a balanced (γ_1, γ_2)-quasi-clique of graph G, let $\gamma_{min} = min\{\gamma_1, \gamma_2\}$, for any $\gamma_1, \gamma_2 \in [0, 1]$, we have:*

$$L(Q) \begin{cases} = 2 & \text{if } \gamma_1 \in [0.5, 1] \wedge \gamma_2 \in [0.5, 1], \\ \leq \frac{3|Q|}{\gamma_{min}(|Q|-1)+1} & \text{others.} \end{cases}$$

Proof. Given a balanced (γ_1, γ_2)-quasi-clique $Q = \{Q_L, Q_R\}$ of graph G, when $\gamma_1 \in [0.5, 1], \gamma_2 \in [0.5, 1]$, let $n = |Q_L|, m = |Q_R|$, for any nodes $u, v \in Q$, there are two cases: (1) if $u, v \in Q_L$ or $u, v \in Q_R$, we have $|N(u) \cup N(v) \cup \{u, v\}| \geq 2(\lceil 0.5(n-1) \rceil + \lceil 0.5m \rceil) + 2 \geq n+m+1 > n+m$; (2) if $u \subseteq Q_L \wedge v \subseteq Q_R$, we have $|N(u) \cup N(v) \cup \{u, v\}| \geq (\lceil 0.5(n-1) \rceil + \lceil 0.5m \rceil + \lceil 0.5(m-1) \rceil + \lceil 0.5n \rceil) + 2 \geq n + m + 1 > n + m$. This means any two nodes in Q must have a common neighbour node. Thus, $L(Q) = 2$.

Next, we prove another case. For each $u \in Q$, we have $d(u) \geq \gamma_{min}(|Q|-1)$. Consider $u, v \in Q$ such that the length of the shortest path between u and v is $l = L(Q)$. Let V_i denote the set of nodes in Q whose shortest distance is exactly i hop away from u, s.t. $\forall w \in V_i \subseteq Q \to dist(u, w) = i$. Then, Q can be partitioned into $(l + 1)$ exclusive groups $V_0, ..., V_l$, s.t., $\bigcup_{0 \leq i \leq l} V_i = Q$, otherwise, $L(Q) > l$.

Then we have: $|V_0| = 1, |V_1| \geq \gamma_{min}(|Q|-1)$ (due to $V_0 = \{u\}$ and $N(u) \subseteq V_1$). Moreover, for any node $w \subseteq V_i (0 < i < l) \to N(w) \subseteq V_{i-1} \cup V_i \cup V_{i+1}$. That means $|V_{i-1} \cup V_i \cup V_{i+1}| = |V_{i-1}| + |V_i| + |V_{i+1}| \geq d(w)+1 \geq \gamma_{min}(|Q|-1)+1$. Therefore, the following series of inequalities are given:

$$\begin{aligned} |V_0| + |V_1| &\geq \gamma_{min}(|Q| - 1) + 1 \\ |V_0| + |V_1| + |V_2| &\geq \gamma_{min}(|Q| - 1) + 1 \\ |V_1| + |V_2| + |V_3| &\geq \gamma_{min}(|Q| - 1) + 1 \\ &\ldots\ldots \\ |V_{l-4}| + |V_{l-3}| + |V_{l-2}| &\geq \gamma_{min}(|Q| - 1) + 1 \\ |V_{l-3}| + |V_{l-2}| + |V_{l-1}| &\geq \gamma_{min}(|Q| - 1) + 1 \\ |V_{l-2}| + |V_{l-1}| + |V_l| &\geq \gamma_{min}(|Q| - 1) + 1 \end{aligned}$$

we sum these inequalities and obtain $3|Q| > l(\gamma_{min}(|Q| - 1) + 1)$. Since $l = L(Q)$, therefore we have:

$$L(Q) < \frac{3|Q|}{\gamma_{min}(|Q| - 1) + 1}$$

Therefore, the theorem is proved. □

B. Proof of Degree-based Pruning Theorems

1) Proof of Theorem 2:

Proof. We hereby prove the pruning rule for $u \in X_L$, and the proof for $v \in X_R$ is symmetric and can be derived similarly.

Consider a valid BQC $Y = \{Y_L, Y_R\} = \{X_L \cup S_L, X_R \cup S_R\}$, where $S_L \subseteq C_L, S_R \subseteq C_R$ and $u \in X_L \subset Y_L$, we have

$$\begin{aligned} d^+_{Y_L}(u) &= d^+_{X_L}(u) + d^+_{S_L}(u) && (10) \\ &< \lceil \gamma_1 \cdot (|X_L| - 1 + d^+_{S_L}(u)) \rceil && (11) \\ &\leq \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil && (12) \end{aligned}$$

$$\begin{aligned} d^-_{Y_R}(u) &= d^-_{X_R}(u) + d^-_{S_R}(u) && (13) \\ &< \lceil \gamma_2 \cdot (|X_R| + d^-_{S_R}(u)) \rceil && (14) \\ &\leq \lceil \gamma_2 \cdot |Y_R| \rceil && (15) \end{aligned}$$

where Eq. (10) and Eq. (13) hold because $Y_L = X_L \cup S_L$ and $Y_R = X_R \cup S_R$; Eq. (11) and Eq. (14) are derived from condition (i) and (ii) respectively, and the fact $S_L \subseteq C_L, S_R \subseteq C_R$ using Lemma 1; Eq. (12) is because $u \in X_L \subset Y_L$ and $X_L \cap S_L = \emptyset$, $|X_L| - 1 + d^+_{S_L}(u) = |X_L \cup N^+_{S_L}(u)| - 1 \leq |X_L \cup S_L| - 1 = |Y_L| - 1$. This result contradicts the fact that $Y = \{Y_L, Y_R\}$ is a balanced $(\gamma_1, \gamma_2)-$quasi-clique. Eq. (15) can be proved similarly. Thus, the condition (i) and (ii) are proved. Node $u \in X_L$ is a failed node for further expansion of X_L, so there is no need to extend X further. □

C. Upper and Lower Bound Derivation

In this section, We define the upper bound (resp. the lower bond) on the number of nodes in $C = \{C_L, C_R\}$ that can be added to X concurrently to form a BQC $Y = \{Y_L, Y_R\}$.

1) Upper Bound Derivation: We take X_L of node set $X = (X_L, X_R)$ as an example, the maximum number of nodes that can be added to X_L to form Y_L is bounded by the minimal positive degree of the nodes in X_L and the minimal negative degree of the nodes in X_R.

So, we first define d^+_{min} (resp. d^-_{min}) as the minimum positive degree (resp. minimum negative degree) of any node in X_L (resp. X_R), where the degrees are calculated based on the interconnections with other nodes in $X_L \cup C_L$.

$$d^+_{minL} = \min_{u \in X_L} \{d^+_{X_L}(u) + d^+_{C_L}(u)\} \quad (16)$$

$$d^-_{minL} = \min_{v \in X_R} \{d^-_{X_L}(v) + d^-_{C_L}(v)\} \quad (17)$$

Now we consider a BQC $Y = \{Y_L, Y_R\}$ such that $X \subseteq Y \subseteq X \cup C$. For any $u \in X_L$, we have $d^+_{X_L}(u) + d^+_{C_L}(u) \geq d^+_{Y_L}(u) \geq \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil \Rightarrow d^+_{minL} \geq \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil$. As a result, $\lfloor d^+_{minL}/\gamma_1 \rfloor \geq \lfloor \gamma_1 \cdot (|Y_L| - 1)/\gamma_1 \rfloor = |Y_L| - 1$, which gives the following upper bound on $|Y_L|$:

$$|Y_L| \leq \lfloor d^+_{minL}/\gamma_1 \rfloor + 1$$

For any $v \in X_R$, we have $d^-_{X_L}(v) + d^-_{C_L}(v) \geq d^-_{Y_L}(v) \geq \lceil \gamma_2 \cdot |Y_L| \rceil \Rightarrow d^-_{minL} \geq \lceil \gamma_2 \cdot |Y_L| \rceil$. Therefore, $\lfloor d^-_{minL}/\gamma_2 \rfloor \geq$

$\lfloor \gamma_2 \cdot |Y_L|/\gamma_2 \rfloor = |Y_L|$, which also gives a upper bound on $|Y_L|$:

$$|Y_L| \leq \lfloor d^-_{minL}/\gamma_2 \rfloor$$

Combining the above two equations, we obtain:

$$|Y_L| \leq min\{\lfloor d^+_{minL}/\gamma_1 \rfloor + 1, \lfloor d^-_{minL}/\gamma_2 \rfloor\}$$

Thus, we could derive $U^{min}_{X_L}$, which is the upper bound on the number of nodes that can be added to X_L to form a valid BQC Y.

$$U^{min}_{X_L} = min\{\lfloor d^+_{minL}/\gamma_1 \rfloor + 1, \lfloor d^-_{minL}/\gamma_2 \rfloor\} - |X_L|$$

We can similarly derive the upper bound on the number of nodes that can be added to $|X_R|$ w.r.t. $U^{min}_{X_R}$:

$$U^{min}_{X_R} = min\{\lfloor d^+_{minR}/\gamma_1 \rfloor + 1, \lfloor d^-_{minR}/\gamma_2 \rfloor\} - |X_R|$$

where d^+_{minR} and d^-_{minR} can be defined similarly in reference to Eq (16) and Eq (17).

Next we introduce lemma 2 to further tighten $U^{min}_{X_L}$.

Basic Lemma for bound derivation. We sort the nodes in $C_L = \{w^+_1, w^+_2, ..., w^+_{|C_L|}\}$ in descending order of their positive degree $d^+_{X_L}(.)$, and also sort the nodes $C_L = \{w^-_1, w^-_2, ..., w^-_{|C_L|}\}$ in descending order of their negative degree $d^-_{X_R}(.)$. Then we have:

Lemma 2. *Given an integer $1 \leq k \leq |C_L|$, if condition (i): $\sum_{u \in X_L} d^+_{X_L}(u) + \sum_{i=1, w^+_i \in C_L}^{k} d^+_{X_L}(w^+_i) < |X_L| \cdot \lceil \gamma_1 \cdot (|X_L| + k - 1) \rceil$, or condition (ii): $\sum_{v \in X_R} d^-_{X_L}(v) + \sum_{i=1, w^-_i \in C_L}^{k} d^-_{X_R}(w^-_i) < |X_R| \cdot \lceil \gamma_2 \cdot (|X_L| + k) \rceil$ holds, then for any node set $S_L \subseteq C_L$ with $|S_L| = k$, there does not exist a valid BQC $Y = (Y_L, Y_R)$ such that $Y_L = X_L \cup S_L$.*

Proof. Consider a valid BQC $Y = \{Y_L, Y_R\}$, s.t., $Y_L = X_L \cup S_L$, $S_L \subseteq C_L$ and $|S_L| = k$, we should have:

$$\begin{aligned}
\sum_{u \in X_L} d^+_{Y_L}(u) &\geq |X_L| \cdot \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil && (18)\\
&= |X_L| \cdot \lceil \gamma_1 \cdot (|X_L| + |S_L| - 1) \rceil && (19)\\
\sum_{v \in X_R} d^-_{Y_L}(v) &\geq |X_R| \cdot \lceil \gamma_2 \cdot |Y_L| \rceil && (20)\\
&= |X_R| \cdot \lceil \gamma_2 \cdot (|X_L| + |S_L|) \rceil && (21)
\end{aligned}$$

However, following condition (i), we have:

$$\begin{aligned}
\sum_{u \in X_L} d^+_{Y_L}(u) &= \sum_{u \in X_L} d^+_{X_L \cup S_L}(u) && (22)\\
&= \sum_{u \in X_L} d^+_{X_L}(u) + \sum_{u \in X_L} d^+_{S_L}(u) && (23)\\
&= \sum_{u \in X_L} d^+_{X_L}(u) + \sum_{w \in S_L} d^+_{X_L}(w) && (24)\\
&\leq \sum_{u \in X_L} d^+_{X_L}(u) + \sum_{i=1, w^+_i \in C_L}^{|S_L|} d^+_{X_L}(w^+_i) && (25)\\
&< |X_L| \cdot \lceil \gamma_1 \cdot (|X_L| + |S_L| - 1) \rceil && (26)
\end{aligned}$$

where Eq. (22) and Eq. (23) hold because $Y_L = X_L \cup S_L$, $S_L \subseteq C_L$ and $C_L \cap X_L = \emptyset$; Eq. (24) is derived because $\sum_{u \in X_L} d^+_{S_L}(u) = \sum_{w \in S_L} d^+_{X_L}(w)$ = number of positive edges connecting S_L and X_L; Eq. (25) holds because $w^+_1, w^+_2, ..., w^+_{S_L}$ are the $|S_L|$ nodes with highest $d^+_{X_L}(.)$ in C_L. Eq. (26) is obtained since condition (i) and $S_L \subseteq C_L \rightarrow |S_L| \leq |C_L|$. Note that Eq. (26) contradicts against Eq. (19), which invalidates Y being a valid BQC. Therefore, condition (i) of this lemma is proved.

Symmetrically, for condition (ii), we have:

$$\begin{aligned}
\sum_{v \in X_R} d^-_{Y_L}(v) &= \sum_{v \in X_R} d^-_{X_L \cup S_L}(v)\\
&= \sum_{v \in X_R} d^-_{X_L}(v) + \sum_{v \in X_R} d^-_{S_L}(v)\\
&= \sum_{v \in X_R} d^-_{X_L}(v) + \sum_{w \in S_L} d^-_{X_R}(w)\\
&\leq \sum_{v \in X_R} d^-_{X_L}(v) + \sum_{i=1, w^-_i \in S_L}^{|S_L|} d^-_{X_R}(w^-_i)\\
&< |X_R| \cdot \lceil \gamma_2 \cdot (|X_L| + |S_L|) \rceil
\end{aligned}$$

which contradicts against Eq. (21), so Y also cannot be a valid BQC under condition (ii).

Therefore, the Lemma 2 holds. □

Upper bound of X_L and X_R. Based on above Lemma 2, we derive a tightened upper bound U_{X_L} as follows:

$$U_{X_L} = max\Bigg\{ t \Bigg| \Big(1 \leq t \leq U^{min}_{X_L}\Big) \bigwedge \Big(\sum_{u \in X_L} d^+_{X_L}(u) + \sum_{i=1, w^+_i \in C_L}^{t} d^+_{X_L}(w^+_i) \geq |X_L| \cdot \lceil \gamma_1 \cdot (|X_L| + t - 1) \rceil \Big) \bigwedge \Big(\sum_{v \in X_R} d^-_{X_L}(v) + \sum_{i=1, w^-_i \in C_L}^{t} d^-_{X_R}(w^-_i) \geq |X_R| \cdot \lceil \gamma_2 \cdot (|X_L| + t) \rceil \Big) \Bigg\}$$

Similarly, we sort $C_R = \{z^+_1, z^+_2, ..., z^+_{|C_R|}\}$ (resp. $C_R = \{z^-_1, z^-_2, ..., z^-_{|C_R|}\}$) in descending order of their positive degree $d^+_{X_R}(.)$ (resp. negative degree $d^-_{X_L}(.)$). Symmetrically, we also derive the tightened upper bound U_{X_R} on the number of nodes that can be added to X_R:

$$U_{X_R} = max\Bigg\{ t \Bigg| \Big(1 \leq t \leq U^{min}_{X_R}\Big) \bigwedge \Big(\sum_{v \in X_R} d^+_{X_R}(v) + \sum_{i=1, z^+_i \in C_R}^{t} d^+_{X_R}(z^+_i) \geq |X_R| \cdot \lceil \gamma_1 \cdot (|X_R| + t - 1) \rceil \Big) \bigwedge \Big(\sum_{u \in X_L} d^-_{X_R}(u) + \sum_{i=1, z^-_i \in C_R}^{t} d^-_{X_L}(z^-_i) \geq |X_L| \cdot \lceil \gamma_2 \cdot (|X_R| + t) \rceil \Big) \Bigg\}$$

If no such t can be found, $U_{X_L} = 0$ (resp. $U_{X_R} = 0$), and X_L (resp. X_R) cannot be extended to form a valid BQC, then the branch $B = (\{X_L, X_R\}, \{C_L, C_R\})$ can be pruned (Type II pruning). Otherwise, if such t exists, based on the derived U_{X_L} and U_{X_R}, we derive upper-bound-based pruning rules in Theorem 3 and Theorem 4.

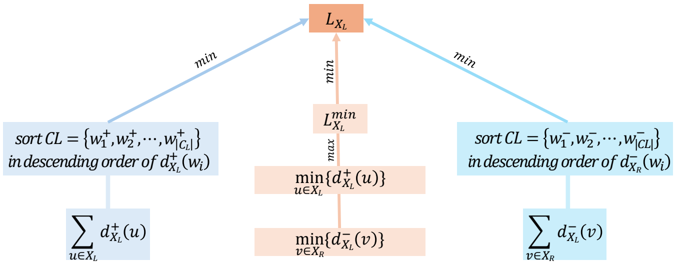

Fig. 15. An overview on the derivation of Lower bounds L_{X_L}. We use A→B to denote A is causing B.

2) Lower Bound Derivation.: Given a node set $X = (X_L, X_R)$, taking X_L as an example, if there exists a node $u \in X_L$ such that $d^+_{X_L}(u) < \lceil \gamma_1 \cdot (|X_L| - 1) \rceil$ or $v \in X_R$ such that $d^-_{X_L}(v) < \lceil \gamma_2 \cdot |X_L| \rceil$, then at least certain number of nodes need to be added to X_L to increase the positive (resp. negative) degree of u in X_L (resp. v in X_R) to form a valid BQC. We denote this lower bound as L_{X_L}, which is defined following the below step-by-step derivation.

We first define $d^+_{X_L}|_{min}$ as the minimum positive degree of any node in X_L and $d^-_{X_L}|_{min}$ as the minimum negative degree of any node in X_R.

$$d^+_{X_L}|_{min} = \min_{u \in X_L} \{d^+_{X_L}(u)\}; \quad d^-_{X_L}|_{min} = \min_{v \in X_R} \{d^-_{X_L}(v)\}$$

Then, the minimum number of nodes needed to be added is derived as follows:

$$L^+_{X_L}|_{min} = min\{t | d^+_{X_L}|_{min} + t \geq \lceil \gamma_1 \cdot (|X_L| + t - 1) \rceil\} \tag{27}$$

$$L^-_{X_L}|_{min} = min\{t | d^-_{X_L}|_{min} + t \geq \lceil \gamma_2 \cdot (|X_L| + t) \rceil\} \tag{28}$$

To satisfy both the above Equations, we obtain:

$$L^{min}_{X_L} = max\{L^+_{X_L}|_{min}, L^-_{X_L}|_{min}\}$$

We can similarly derive the lower bound on the number of nodes that can be added to $|X_R|$ w.r.t. $L^{min}_{X_R}$:

$$L^{min}_{X_R} = max\{L^+_{X_R}|_{min}, L^-_{X_R}|_{min}\}$$

where $L^+_{X_R}|_{min}$ and $L^-_{X_R}|_{min}$ can be derived similarly in reference to Eq (27) and Eq (28).

Lower bound of X_L and X_R. Based on Lemma 2, we define a tightened lower bound L_{X_L} as follows:

$$L_{X_L} = min\Bigg\{t \Bigg| \Big(L^{min}_{X_L} \leq t \leq |C_L|\Big) \bigwedge \Big(\sum_{u \in X_L} d^+_{X_L}(u) + \sum_{i=1, w_i^+ \in C_L}^{t} d^+_{X_L}(w_i^+) \geq |X_L| \cdot \lceil \gamma_1 \cdot (|X_L| + t - 1) \rceil\Big) \bigwedge \Big(\sum_{v \in X_R} d^-_{X_L}(v) + \sum_{i=1, w_i^- \in C_L}^{t} d^-_{X_R}(w_i^-) \geq |X_R| \cdot \lceil \gamma_2 \cdot (|X_L| + t) \rceil\Big)\Bigg\}$$

Similarly, we can derive the tightened lower bound on the number of nodes that can be added to X_R, w.r.t. L_{X_R}:

$$L_{X_R} = min\Bigg\{t \Bigg| \Big(L^{min}_{X_R} \leq t \leq |C_R|\Big) \bigwedge \Big(\sum_{v \in X_R} d^+_{X_R}(v) + \sum_{i=1, z_i^+ \in C_R}^{t} d^+_{X_R}(z_i^+) \geq |X_R| \cdot \lceil \gamma_1 \cdot (|X_R| + t - 1) \rceil\Big) \bigwedge \Big(\sum_{u \in X_L} d^-_{X_R}(u) + \sum_{i=1, z_i^- \in C_R}^{t} d^-_{X_L}(z_i^-) \geq |X_L| \cdot \lceil \gamma_2 \cdot (|X_R| + t) \rceil\Big)\Bigg\}$$

If no such t exists, X_L or X_R cannot be extended to form a valid BQC, then the branch $B = (\{X_L, X_R\}, \{C_L, C_R\})$ can be pruned (Type II pruning). Otherwise, if such t exists, based on the derived L_{X_L} and L_{X_R}, we derive lower-bound-based pruning rules in Theorem 5 and Theorem 6.

D. Proof of Bound-based Pruning Theorems

1) Proof of Theorem 3:

Proof. We hereby prove the pruning rule for $u \in C_L$, and the proof for $v \in C_R$ is symmetric and can be similarly derived.

Consider a BQC $Y = \{Y_L, Y_R\} = \{X_L \cup S_L, X_R \cup S_R\}$ where $u \in S_L \subseteq C_L$, if condition (i) holds, we have $d^+_{Y_L}(u) = d^+_{X_L}(u) + d^+_{S_L}(u)$ (since $Y_L = X_L \cup C_L$ and $X_L \cap C_L = \emptyset$) $\leq d^+_{X_L}(u) + |S_L| - 1$ (since $u \in S_L$) $< \lceil \gamma_1 \cdot (|X_L| + |S_L| - 1) \rceil$ (due to condition (i) and $|S_L| \leq U_{X_L}$ using Lemma 1) $= \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil$. This contradicts the fact that Y is a BQC. Similarly, if condition (ii) holds, we have $d^-_{Y_R}(u) = d^-_{X_R}(u) + d^-_{S_R}(u) \leq d^-_{X_R}(u) + |S_R| < \lceil \gamma_2 \cdot (|X_R| + |S_R|) \rceil = \lceil \gamma_2 \cdot |Y_R| \rceil$, which also contradicts the fact that Y is a BQC. Therefore, if u satisfies either condition (i) or (ii), u is not the valid candidate node and can be pruned from C_L. □

2) Proof of Theorem 4:

Proof. We hereby prove the pruning rule for $u \in X_L$, and the proof for $v \in X_R$ is symmetric and can be derived similarly.

Consider a valid BQC $Y = (Y_L, Y_R) = (X_L \cup S_L, X_R \cup S_R)$ where $u \in X_L$, if condition (i) holds, we have: $d^+_{Y_L}(u) = d^+_{X_L}(u) + d^+_{S_L}(u) \leq d^+_{X_L}(u) + |S_L|$ (since $u \in X_L$ and $X_L \cap S_L = \emptyset$) $< \lceil \gamma_1 \cdot (|X_L| + |S_L| - 1) \rceil$ (due to condition (i) and $|S_L| \leq U_{X_L}$ using Lemma 1) $= \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil$. This contradicts the fact that Y is a BQC. Similarly, if condition (ii) holds, we have: $d^-_{Y_R}(u) = d^-_{X_R}(u) + d^-_{S_R}(u) \leq d^-_{X_R}(u) + |S_R| < \lceil \gamma_2 \cdot (|X_R| + |S_R|) \rceil = \lceil \gamma_2 \cdot |Y_R| \rceil$, which also contradicts the fact that Y is a BQC. Thus, if u satisfies either condition (i) or (ii), there does not exist a BQC $Y = (Y_L, Y_R)$, s.t., $u \in X_L \subseteq Y_L$. And $u \in X_L$ is a failed node further expansion of X_L, so there is no need to extend X further. □

3) Proof of Theorem 5:

Proof. We prove the pruning rule for $u \in C_L$, and another pruning rule for $v \in C_R$ can be proved symmetrically.

Consider a BQC $Y = \{Y_L, Y_R\} = \{X_L \cup S_L, X_R \cup S_R\}$ where $u \in S_L \subseteq C_L$ and $S_R \subseteq C_R$, if condition (i) holds, we have: $d^+_{Y_L}(u) = d^+_{X_L}(u) + d^+_{S_L}(u) \leq d^+_{X_L}(u) + d^+_{C_L}(u) < \lceil \gamma_1 \cdot (|X_L| + L_{X_L} - 1) \rceil$ (due to condition (i)) $\leq \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil$

(since $L_{X_L} \leq |S_L|$ and $Y_L = X_L \cup S_L$), which contradicts the fact that Y is a BQC. Similarly, if condition (ii) holds, we have: $d^-_{Y_R}(u) = d^-_{X_R}(u) + d^-_{S_R}(u) \leq d^-_{X_R}(u) + d^-_{C_R}(u) < \lceil \gamma_2 \cdot (|X_R| + L_{X_R}) \rceil$ (due to the condition (ii)) $\leq \lceil \gamma_2 \cdot |Y_R| \rceil$ (since $L_{X_R} \leq |S_R|$ and $Y_R = X_R \cup S_R$), which also contradicts the fact that Y is a BQC.

To conclude, if u satisfies either Condition (i) or (ii), u is not the valid candidate node and can be pruned from C_L. The aforementioned conditions (i) and (ii) hold. □

4) Proof of Theorem 6:

Proof. We hereby prove the pruning rule of $u \in X_L$, and the proof for $v \in X_R$ is symmetric and can be proved similarly.

Consider a valid BQC $Y = (Y_L, Y_R) = (X_L \cup S_L, X_R \cup S_R)$ where $S_L \subseteq C_L$ and $S_R \subseteq C_R$, for $u \in X_L$, if condition (i) holds, we have: $d^+_{Y_L}(u) = d^+_{X_L}(u) + d^+_{S_L}(u) \leq d^+_{X_L}(u) + d^+_{C_L}(u)$ (since $S_L \subseteq C_L$) $< \lceil \gamma_1 \cdot (|X_L| + L_{X_L} - 1) \rceil$ (due to condition (i)) $\leq \lceil \gamma_1 \cdot (|Y_L| - 1) \rceil$ (since $L_{X_L} \leq S_L$ and $Y_L = X_L \cup S_L$), which contradicts the fact that Y is a BQC. Similarly, we have $d^-_{Y_R}(u) = d^-_{X_R}(u) + d^-_{S_R}(u) \leq d^-_{X_R}(u) + d^-_{C_R}(u) < \lceil \gamma_2 \cdot (|X_R| + L_{X_R}) \rceil \leq \lceil \gamma_2 \cdot |Y_R| \rceil$, which also contradicts the fact that Y is a BQC.

To conclude, if u satisfies either Condition (i) or (ii), there does not exist a BQC $Y = (Y_L, Y_R)$, s.t., $u \in X_L \subseteq Y_L$, the aforementioned conditions (i) or (ii) hold. This means $u \in X_L$ is a failed node for extension, so there is no need to further expand X. □

Appendix B
Additional algorithms

A. Positive and Negative Candidates Computation

Algorithm 3: ComputePCNC(G, u, l)

Input: Signed Graph G, node u, diameter bound L
Output: $PC(u)$, $NC(u)$

1 $PC^1(u) \leftarrow N^+(u)$; $\quad NC^1(u) \leftarrow N^-(u)$
2 $PC(u) \leftarrow PC^1(u)$; $\quad NC(u) \leftarrow NC^1(u)$
3 **for** $l \in [2, L]$ **do**
4   **foreach** $w \in PC^{l-1}(u)$ **do**
5         $PC^l(u) \leftarrow PC^l(u) \cup N^+(w)$
6         $NC^l(u) \leftarrow NC^l(u) \cup N^-(w)$
7   **foreach** $w \in NC^{l-1}(u)$ **do**
8         $PC^l(u) \leftarrow PC^l(u) \cup N^-(w)$
9         $NC^l(u) \leftarrow NC^l(u) \cup N^+(w)$
10     $PC(u) \leftarrow PC(u) \cup PC^l(u)$
11     $NC(u) \leftarrow NC(u) \cup NC^l(u)$
12 tmp $\leftarrow PC(u) \cap NC(u)$
13 $PC(u) \leftarrow PC(u) -$ tmp, $\quad NC(u) \leftarrow NC(u) -$ tmp
14 **return** $PC(u), NC(u)$

Algorithm 3 is designed to calculate the positive/negative candidate sets of a given node u. We first initialize $PC^1(u) = N^+(u)$, $NC^1(u) = N^-(u)$ according to Eq. (1) (line 1). Then, lines 2-11 recursively compute $PC^l(u)$ and $NC^l(u)$ and merge them to obtain the positive/negative candidates $PC(u)$ and $NC(u)$ according to Eq. (2) and Eq. (3). We further shrink $PC(u)$ and $NC(u)$ (line 12-13) by pruning the invalid node v, s.t., $v \in PC(u) \wedge v \in NC(u)$. This means v are both positive and negative candidates of u, which contradicts structural balance theory, so v is an invalid candidate and should be removed.

B. BQC Validation and Look-Ahead Technique

BQC Validation. Algorithm 4 verify if the given node sets $G[X_L, X_R]$ is a valid BQC. It first validates that X_L and X_R adheres to the required size threshold (lines 1-2). Then, it examines each node $u \in X_L$ to ascertain whether its positive and negative degrees satisfy specified degree constraints (lines 3-5). An analogous verification process is applied to each node $v \in X_R$ (lines 6-8). If every node in (X_L, X_R) passes all validation checks, then $G[X_L, X_R]$ is deemed a valid BQC, prompting the algorithm to return *True* (line 9). Conversely, should any node fail to satisfy the requisite constraints, the algorithm aborts the process and returns *False* (lines 2, 5, 8).

Algorithm 4: BQC_verify($X_L, X_R, \gamma_1, \gamma_2, k$)

1 **if** $|X_L| < k \vee |X_R| < k$ **then**
2   **return** *False*
3 **for** *each* $u \in X_L$ **do**
4   **if** $d^+_{X_L}(u) < \lceil \gamma_1 \cdot (|X_L| - 1) \rceil \vee d^-_{X_R}(u) < \lceil \gamma_2 \cdot |X_R| \rceil$
5   **return** *False*
6 **for** *each* $v \in X_R$ **do**
7   **if** $d^+_{X_R}(v) < \lceil \gamma_1 \cdot (|X_R| - 1) \rceil \vee d^-_{X_L}(v) < \lceil \gamma_2 \cdot |X_L| \rceil$
8   **return** *False*
9 **return** *True*

Look-ahead Technique. This technique checks whether $G[X_L \cup C_L, X_R \cup C_R]$ is a valid BQC before expanding X. If so, there is no need to further expand X_L or X_R, avoiding unnecessary computations. In implementation, we specify $X_L \cup C_L$ and $X_R \cup C_R$ as inputs to Algorithm 4 to verify whether the induced graph $G[X_L \cup C_L, X_R \cup C_R]$ satisfies the conditions for being a valid BQC.

C. Branch Pruning based on Degree/Bound/Critical-node

Iterative Branch Pruning. We use degree-based, bound-based and critical node based pruning rules to iteratively prune unpromising search spaces in a branch $B = \langle \{X_L, X_R\}, \{C_L, C_R\}, \{D_L, D_R\} \rangle$. When a failed candidate node is removed from $\{C_L, C_R\}$ and/or a valid candidate node is added to $\{X_L, X_R\}$, the degrees of nodes in branch with respect to X_L, X_R, C_L, C_R would be accordingly updated, and the bounds $L_{X_L}, U_{X_L}, L_{X_R}, U_{X_R}$ would be accordingly updated (Appendix A-C), potentially creating new possibility for degree-based pruning (Section IV-A) and bound-based pruning (Section IV-B). The pruning is iteratively carried out until no more invalid nodes can be removed.

Algorithm 5 shows the process of iterative branch pruning. We first compute $U_{X_L}, L_{X_L}, U_{X_R}, L_{X_R}$ based on Appendix A-C (line 1). If $L_{X_L} > U_{X_L}$ or $L_{X_R} > U_{X_R}$ is satisfied

Algorithm 5: branch_pruning($X_L, X_R, C_L, C_R, D_L, D_R$)

1 compute bounds $U_{X_L}, L_{X_L}, U_{X_R}, L_{X_R}$
2 **if** $L_{X_L} > U_{X_L}$ **then** $X_L \leftarrow \emptyset$
3 **if** $L_{X_R} > U_{X_R}$ **then** $X_R \leftarrow \emptyset$
4 **if** *there is a critical node* $u \in X_L$ *or* $v \in X_R$ **then**
5   add its positive/negative neighbors to corresponding X_L or X_R and update C_L, C_R, D_L, D_R
6   update $U_{X_L}, L_{X_L}, U_{X_R}, L_{X_R}$
7 **if** $L_{X_L} \leq U_{X_L}$ *or* $L_{X_R} \leq U_{X_R}$ **then**
8   identify failed nodes in $X_L \cup X_R$ using degree, bound-based Type II pruning rules
9   **if** *any failed node exist in* $X_L \cup X_R$ **then**
10   **return** *branch_pruned*
11   filter invalid nodes in $C_L \cup C_R$ using degree-based and bound-based Type I pruning rules
12   **while** $X_L \cup X_R \neq \emptyset$ *and* $C_L \vee C_R$ *shrank* **do**
13   update $U_{X_L}, L_{X_L}, U_{X_R}, L_{X_R}$
14   identify failed nodes in $X_L \cup X_R$ based on Type II pruning rules
15   **if** *any failed node exist in* $X_L \cup X_R$ **then**
16   **return** *branch_pruned*
17   filter invalid candidate nodes in $C_L \cup C_R$ using Type I degree pruning rule
18 **else**
19     $X_L \leftarrow \emptyset$; $X_R \leftarrow \emptyset$
20 **return** *branch_not_pruned*

(lines 2-3), it means that X_L or X_R cannot be expanded further into a valid BQC, we set X_L, X_R as empty set. Otherwise, it will enter the iterative pruning procedure (lines 7-19). We first identifies critical nodes in X_L or X_R and if there exists such nodes (lines 4-6), the algorithm adds all its neighbors in C_L or C_R into X_L or X_R based on Theorem 7. Next, it checks the extensibility of the nodes in $X_L \cup X_R$ based on Type II pruning rules (line 8). And if any failed node exists (line 9), the algorithm return tag *branch_pruned* (line 10) so that the main enumeration algorithm will skip this branch. Then, the algorithm filter out invalid candidate nodes in $C_L \cup C_R$ based on Type I pruning rules (line 11). If $X_L \cup X_R$ is not empty and some nodes have been pruned from $C_L \cup C_R$ (line 12), then the degree of nodes within branch may change, triggering the update of $U_{X_L}, L_{X_L}, U_{X_R}, L_{X_R}$ (line 13) and creating more Type I and Type II pruning opportunities (lines 14-17). If the branch is not pruned after all pruning rules check, it finally return *branch_not_pruned* (line 20).