A OMITTED PROOFS FOR INDEX COMPRESSIONS

Proof of theorem 4.6. The approximation B' is unbiased because each butterfly from T_E that contributes to B is preserved with probability λ_1, and each of these preserved butterfly contributes $1/\lambda_1$ to B'.

B' is the sum of independent random variables taking values in $\{0, 1/\lambda_1\}$. By Chernoff bound, we have

$$\mathbb{P}\left[|B' - \mathbb{E}\left[B'\right]| \geq \delta\mathbb{E}\left[B'\right]\right] = O(\exp(-\delta^2\lambda_1\mathbb{E}\left[B'\right]/3)).$$

Since B' is unbiased, $\mathbb{E}\left[B'\right] = B$. Substituting $\delta = c\log^{0.5}(n)\lambda_1^{-0.5}B^{-0.5}$ for some large enough c into the equation above gives

$$\begin{aligned}&\mathbb{P}\left[|B' - B| \geq \delta B\right]\\ =&O(\exp(-\delta^2\lambda_1 B/3))\\ =&O(\exp(-c^2\log(n))) = O(n^{-c^2}).\end{aligned}$$

□

Proof of theorem 4.7. We consider any two different wedges w_1, w_2 from the same $\mathcal{CS} \in T_C$ such that both appears in the projected graph $G_{[t_s,t_e]}$ for the time-window $[t_s, t_e]$. Let $\widetilde{\mathcal{CS}}$ be the corresponding data structure of $\mathcal{CS}$ in $\widetilde{T_C}$. Let C' be the number of wedges in $G_{[t_s,t_e]}$ that are inserted into $\widetilde{\mathcal{CS}}$. Let C be the total number of wedges in $G_{[t_s,t_e]}$ that are inserted into $\mathcal{CS}$. The pair of wedges, (w_1, w_2), contributes $1/\lambda_2^2$ to the answer $\binom{C'}{2}/\lambda_2^2$ if both of them are preserved $\widetilde{\mathcal{CS}}$. The expected contribution is exactly 1 because w_1 and w_2 are preserved independently with probability λ_2 each. Thus, by the linearity of expectation, $\mathbb{E}\left[\binom{C'}{2}/\lambda_2^2\right]$ is exactly $\binom{C}{2}$, i.e., the number of pairs (w_1, w_2). Summing over all $\mathcal{CS} \in T_C$ gives the unbiasedness.

Let δ be an undetermined coefficient. By Chernoff bound,

$$\begin{aligned}&\mathbb{P}\left[|C' - \lambda_2 C| \geq \delta C\right]\\ \leq&O(\exp(-\Omega(-\delta^2\lambda_2 C))).\end{aligned}$$

Let $\delta = \log^{0.5}(n)\lambda_2^{-0.5}C^{-0.5}$. We have that with high probability, $|C' - \lambda_2 C| = O(C^{0.5}\log^{0.5}(n)\lambda_2^{-0.5})$. This means that $\binom{C'}{2}/\lambda_2^2$ is within multiplicative error $O(C^{-0.5}\lambda_2^{-1.5}\log^{0.5}(n))$ of the correct answer $\binom{C}{2}$.

Conditioning on the error bound above hold for every $\mathcal{CS}$ in T_C, we may use Hoeffding's inequality. For each $\mathcal{CS}_i \in T_C$ such that $|\mathcal{CS}_i| \geq 2$, $\binom{C_i'}{2}/\lambda_2^2$ is an independent random variable taking values from

$$\binom{C_i}{2} \pm O(C^{1.5}\lambda_2^{-3.5}\log^{0.5}(n)),$$

where C_i and C_i' are the number of wedges in $\mathcal{CS}_i \cap G_{[t_s,t_e]}$ and the number of preserved wedges in $\mathcal{CS}_i \cap G_{[t_s,t_e]}$. Let $S' = \sum_i \binom{C_i'}{2}/\lambda_2^2$, and let $S = \sum_i \binom{C_i}{2}$. We have, for any parameter t,

$$\begin{aligned}&\mathbb{P}\left[|S' - S| \geq t\right]\\ \leq&O\left(\exp\left(-\Omega\left(t^2/\left(\lambda_2^{-3.5}\log^{0.5}(n)\sum_i C_i^3\right)\right)\right)\right)\end{aligned}$$

by Hoeffding's inequality. We may set $t = c\lambda_2^{-1.75}\log^{0.75}(n)S^{0.75}$ for large enough constant c to ensure that $|S' - S| < t$ with high probability.

□

B HANDLING DUPLICATE EDGES

The key challenge when the graph has duplicate edges is that the life cycle of a wedge or a butterfly is no longer an active timestamp as defined in Definition 4.2. As we will see in our technical report Part B, the life cycle can be decomposed into several redefined active timestamps (Definition B.1) for graphs with duplicate edges. We will prove that the decomposition does not increase the time complexity or memory usage of GSI (Lemma B.1 and Theorem B.2).

Definition B.1 (Active Intervals for Graphs with Duplicate Edges). *Given a bipartite temporal graph G with duplicate edges, a subgraph P, we define the active intervals $\tilde{\mathcal{T}}(P)$ as a tuple of timestamps of the form $[l, r_1, r_2]$ $(l \leq r_1 \leq r_2)$ such that P is active in the query time-window $[t_s, t_e]$ if and only if for exactly one of the timestamps $[l, r_1, r_2]$, $t_s \leq l$ and $r_1 \leq t_e < r_2$.*

For applying GSI to a graph with duplicate edges, we first modify $\mathcal{CS}$ such that it can answer 2D-range queries on timestamps of the form $[l, r_1, r_2]$, i.e., counting the number of $[l, r_1, r_2]$ such that $t_s \leq l$ and $r_1 \leq t_e \leq r_2$ given the query time-window $[t_s, t_e]$. This can be done by applying the inclusive-exclusive principle on 2D ranges. Then we generate the active intervals (tuple of timestamps $[l, r_1, r_2]$) for each wedge. We feed each $[l, r_1, r_2]$ to GSI (with the modified $\mathcal{CS}$).

Now, we provide the intuition and detailed analysis of Algorithm 7.

Given a butterfly or wedge P, we consider a subgraph S constructed by choosing exactly one edge (u, v, t) for each $(u, v) \in E(P)$. We denote the set of all possible Ss by $\mathcal{S}$. Due to duplicate edges, $|\mathcal{S}| > 1$. For each $S \in \mathcal{S}$, its active timestamp is $[l_S, r_S]$ where $l_S = \min_{(u,v,t)\in S} t$ and $r_S = \max_{(u,v,t)\in S} t$. For a time-window $[t_s, t_e]$, $P \in G_{[t_s,t_e]}$ if and only if there exists an $S \in \mathcal{S}$ satisfying $t_s \leq l_S \leq r_S \leq t_e$. Therefore, for $S_1, S_2 \in \mathcal{S}$, if $l_{S_1} \leq l_{S_2} \leq r_{S_2} \leq r_{S_1}$, then S_1 is not necessary to be considered for any query time-window. In this way, we manage to reduce the number of Ss to be stored to answer historical queries concerning P.

We still need to resolve the issue of overcounting. For a time-window $[t_s, t_e]$, if there are multiple Ss such that their timestamps are all included in $[t_s, t_e]$, we should not count P as multiple wedges or butterflies (Definition 3.3). This is the reason we consider the redefined active timestamp in Definition B.1. Let the reduced set of Ss be $\{[l_1, r_1], [l_2, r_2], \cdots, [l_k, r_k]\}$ satisfying $l_i < l_{i+1}, r_i \leq r_{i+1}$ for each $1 \leq i < k$. We create a redefined timestamp $[l_i, r_i, r_{i+1}]$ for each $1 \leq i < k$ and a redefined timestamp $[l_k, r_k, \infty]$ for the last timestamp $[l_k, r_k]$. We can see that exactly one of the redefined timestamps becomes active when $P \in G_{[t_s,t_e]}$: If there exists $1 \leq i < k$ such that $r_i \leq t_e < r_{i+1}$, only $[l_i, r_i, r_{i+1}]$ is active. Otherwise, we have $t_e \geq r_k$. In such case, $[l_k, r_k, \infty]$ is active. In addition, if $P \notin G_{[t_s,t_e]}$, no redefined timestamps becomes active. For any timestamp $[l_i, r_i]$, $[t_s, t_e]$ does not include $[l_i, r_i]$. This implies that the redefined timestamp $[l_i, r_i, r_{i+1}]$ (or $[l_k, r_k, \infty]$ when $i = k$) is not active.

Algorithm 7 computes the active intervals for a wedge $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$. Given the two timestamp sets $L_{x,y}$ and $L_{y,z}$, we will produce all necessary $S \in \mathcal{S}$ and store the redefined timestamps in A. We can safely assume that there is no duplicate timestamp in either $L_{x,y}$ or $L_{y,z}$. To begin with, we initialize S to be $\emptyset$ (Line 3). We sort $L_{x,y}$ and $L_{y,z}$ from small to large (Line 2). We also insert an

Algorithm 7: Computing $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$'s Active Timestamps

```
Input: A list of unique timestamps on the duplicate edges
       between $x$ and $y$: $L_{x,y}$; A list of unique timestamps
       on the duplicate edges between $y$ and $z$ : $L_{y,z}$;
Output: The active intervals $A = \{[l_i, r_{1,i}, r_{2,i}]\}$ for
        $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$
1  $L_{x,y} \leftarrow L_{x,y} \cup \{\infty\}, L_{y,z} \leftarrow L_{y,z} \cup \{\infty\}$;
2  Sort $L_{x,y}$ and $L_{y,z}$ in ascending order;
3  $S \leftarrow \emptyset, i \leftarrow 1, j \leftarrow 1$;
4  if $L_{x,y}[1] \leq L_{y,z}[1]$ then
5      $i \leftarrow$ the maximum $k$ such that $L_{x,y}[k] \leq L_{y,z}[1]$;
6  else
7      $j \leftarrow$ the maximum $k$ such that $L_{y,z}[k] \leq L_{x,y}[1]$;
8  $l \leftarrow \infty, r \leftarrow \infty$;
9  while $i < |L_{x,y}|$ and $j < |L_{y,z}|$ do
10     $l' \leftarrow \min(L_{x,y}[i], L_{y,z}[j]), r' \leftarrow \max(L_{x,y}[i], L_{y,z}[j])$;
11     if $r \neq \infty$ then $A = A \cup \{[l, r, r']\}$ ;
12     if $L_{x,y}[i] \leq L_{y,z}[j]$ then
13         $i \leftarrow i + 1$;
14         $j \leftarrow$ the maximum $k$ such that $L_{y,z}[k] \leq L_{x,y}[i]$;
15     else
16         $j \leftarrow j + 1$;
17         $i \leftarrow$ the maximum $k$ such that $L_{x,y}[k] \leq L_{y,z}[j]$;
18     $l \leftarrow l', r \leftarrow r'$;
19 if $r \neq \infty$ then
20     $A = A \cup \{[l, r, \infty]\}$
21 return $S$;
```

∞ timestamp to both timestamp sets (Line 1) to avoid boundary cases. We will enumerate every element in these sets, and we denote them as $L_{x,y}[i](1 \leq i \leq |L_{x,y}|$ and $L_{y,z}[j](1 \leq j \leq |L_{y,z}|$. Consider a pair (i, j). When $L_{x,y}[i] \leq L_{y,z}[j]$, we will adjust i to i' such that $L_{x,y}[i+1] > L_{y,z}[j]$. That is to say, we will adjust the smaller side as much as possible without breaking the inequality $L_{x,y}[i] \leq L_{y,z}[j]$ (Line 5, Line 14, Line 7, and Line 17). We do the same for the case which $L_{x,y}[i] > L_{y,z}[j]$ (adjusting j instead). For the current enumerated pair (i, j), the corresponding redefined timestamp of the wedge formed by $(x, y, L_{x,y}[i])$ and $(y, z, L_{y,z}[j])$ is $[l, r, \max(L_{x,y}[i], L_{y,z}[j])]$, where l and r are the minima and maxima of the two timestamps $L_{x,y}[i]$s and $L_{y,z}[j]$s in the previous iteration (Line 10 and Line 11). We insert it into A (Line 11). After inserting, we move to the next pair by increasing one of the indices i or j. When $L_{x,y}[i] \leq L_{y,z}[j]$, by setting i to $i + 1$, we will have $L_{x,y}[i] > L_{y,z}[j]$ (Line 12 to Line 14). Then we adjust j again to the largest possible value satisfying $L_{y,z}[j] \leq L_{x,y}[i]$ (Line 14). If $L_{x,y}[i] > L_{y,z}[j]$, we increase j and adjust i instead (Line 13 and Line 17). The whole process will terminate when no more feasible pair needs to be enumerated (Line 9). Then, all the redefined timestamps are stored in A. We return A as the output (Line 21).

Lastly, we prove that counting the number of activated redefined timestamps can be converted to computing the difference between two 2D-range counting queries.

Previously, we regarded the active timestamp as a single point in the 2-D plane and inserted it into the 2D-counting data structure. For any time window, the answer is computed by a single query on the data structure. Under the current definition of the active timestamp, we can still efficiently answer the query using two 2D-range counting data structures T and $\overline{T}$ instead. That is to say, after construction, we will be able to answer the number of redefined timestamps $[l, r_1, r_2]$ satisfying $t_s \leq l$ and $r_1 \leq t_e < r_2$. Specifically, for $[l, r_1, r_2]$, we insert (l, r_1) and (l, r_2) into T and $\overline{T}$ respectively. To count the number of active timestamps of a time-window $[t_s, t_e]$, we first query $[t_s, \infty] \times [-\infty, t_e]$ on both T and $\overline{T}$, denoted as *num* and $\overline{num}$ respectively. Then, we return $num - \overline{num}$ as the answer. The formal proof is as follows.

$$\sum_{P \in G} \sum_{[l_i, r_{1,i}, r_{2,i}] \in \tilde{\mathcal{T}}(P)} \mathbb{1}\{l_i \geq t_s \wedge t_e \in [r_{1,i}, r_{2,i})\} =$$

$$\sum_{P \in G} \sum_{[l_i, r_{1,i}, r_{2,i}] \in \tilde{\mathcal{T}}(P)} \mathbb{1}\{l_i \geq t_s \wedge t_e \geq r_{1,i}\} - \mathbb{1}\{l_i \geq t_s \wedge t_e \geq r_{2,i}\} =$$

$$\left(\sum_{P \in G} \sum_{[l_i, r_{1,i}] \in \tilde{\mathcal{T}}(P)} \mathbb{1}\{l_i \geq t_s \wedge t_e \geq r_{1,i}\} \right) -$$

$$\left(\sum_{P \in G} \sum_{[l_i, r_{2,i}] \in \tilde{\mathcal{T}}(P)} \mathbb{1}\{l_i \geq t_s \wedge t_e \geq r_{2,i}\} \right) =$$

$$T.query([t_s, \infty] \times [-\infty, t_e]) - \overline{T}.query([t_s, \infty] \times [-\infty, t_e])$$

Lastly, we prove that if the size (number of $[l, r_1, r_2]$ in the tuple) of the active intervals of each wedge is bounded (Lemma B.1), both EBI and CBI's time complexity will not be compromised.

Lemma B.1. *For any two vertices $u, v \in V(G)$, we denote $cnt_{u,v}$ as the number of edges $(u, v, t) \in E(G)$. There exists an algorithm (Algorithm 7 in the technical report Part B) that returns its active intervals (Definition B.1) of size $O(\min(cnt_{x,y}, cnt_{y,z}))$ for any wedge $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$.*

Proof of Lemma B.1. WLOG, we assume $cnt_{x,y} \leq cnt_{y,z}$. It is easy to see the time complexity of Algorithm 7 is $O(|S| \log m)$, and we will then show $|S| \leq 3cnt_{x,y}$.

For each wedge $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$ $[l_i, r_{1,i}, r_{2,i}] \in S$, we have at least one of l_i or $r_{1,i}$ comes from a timestamp of $L_{x,y}$. In other words, for each pair (l, r) in lines 9 to 18 of Algorithm 7, there exists an $i \in [1, cnt_{x,y}]$ such that at least one of l or r equals to $L_{x,y}[i]$. We consider these cases as follows:

- $l = L_{x,y}[i], r \neq L_{x,y}[i]$: We move $i \leftarrow i + 1$ in the next iteration, which implies this case will only occur for at most $cnt_{x,y}$ times.
- $l \neq L_{x,y}[i], r = L_{x,y}[i]$: let $\hat{l}$ and $\hat{r}$ be the values of l and r in the next iteration, respectively. We have $\hat{r} \geq \hat{l} \geq r$, which implies this case will only occur for at most $cnt_{x,y}$ times.
- $l = L_{x,y}[i], r = L_{x,y}[i]$: let $\hat{l}$ and $\hat{r}$ be the value of l and r in the next iteration, respectively. We have either $r' \geq \hat{l} > r$ or $\hat{r} > \hat{l} \geq r$, which implies this case will only occur for at most $cnt_{x,y}$ times.

Since each wedge belongs to one of the three cases above, we prove that $|S| \leq 3cnt_{x,y}$.

□

With this lemma, we are ready to prove that the time complexity for our algorithms will not be compromised by duplicate edges:

Theorem B.2 (Time complexity with Duplicate Edges). ***(i)*** *There exists a modification for* `EBI` *that can run in* $O(\log m)$ *time and* $O(m^2)$ *memory usage for bipartite temporal graphs with duplicate edges;* ***(ii)*** *There exists a modification for* `CBI` *that can run in* $O(\tilde{w} \log m)$ *time and* $O(m\delta)$ *memory usage for bipartite temporal graphs with duplicate edges.*

Proof of Theorem B.2. **(i)**: Considering a butterfly consisting of two wedges $\langle x \rightsquigarrow y_1 \rightsquigarrow z\rangle$ and $\langle x \rightsquigarrow y_2 \rightsquigarrow z\rangle$, the size of its active intervals is $O(\min(cnt_{x,y_1}, cnt_{y_1,z}) \times \min(cnt_{x,y_2}, cnt_{y_2,z}))$ by Lemma B.1. Since

$$\begin{aligned}
&\sum_{\text{each butterfly } \langle x,y_1,z,y_2\rangle \in G} \min(cnt_{x,y_1}, cnt_{y_1,z}) \times \min(cnt_{x,y_2}, cnt_{y_2,z}) \\
\leq &\sum_{\text{each butterfly } \langle x,y_1,z,y_2\rangle \in G} cnt_{x,y_1} cnt_{y_2,z} \leq m^2.
\end{aligned}$$

, the bound of the number of points in $\mathcal{CS}s$ $O(m^2)$ for `EBI` still holds.

(ii): Considering a wedge $\langle x \rightsquigarrow y \rightsquigarrow z\rangle$, the the size of its active intervals in its active timestamp is $O(\min(cnt_{x,y}, cnt_{y,z}))$ by Lemma B.1. The total number of tuples for all wedges' active timestamps is:

$$\begin{aligned}
&\sum_{y \in V(G)} \sum_{\substack{x \in N(y), \\ pr(x) \prec pr(y)}} \sum_{\substack{z \in N(y), \\ pr(x) \prec pr(z)}} \min(cnt_{x,y}, cnt_{y,z}) \\
\leq &\sum_{y \in V(G)} \sum_{\substack{x \in N(y), \\ pr(x) \prec pr(y)}} cnt_{x,y} deg_x \\
\leq &\sum_{(x,y) \in E(G)} \min(deg_x, deg_y) = m\delta.
\end{aligned}$$

Therefore, the bound of the number of points in $\mathcal{CS}s$ for `CBI`, $O(m\delta)$, also holds. □

C OMITTED PROOF FOR POWER-LAW GRAPHS

Proof for Theorem 5.1 and Theorem 5.2. By Theorem 4.5, we know that the query time for `GSI` is nearly linear in the number of keys (x, z) in T_C. The space usage for `GSI`is bounded by the number of butterflies not maintained by $T_C[(x, z)]$, plus the total number of wedges in each $W[(x, z)]$ for $(x, z) \in T_C.keys()$.

To bound the query time and space usage, let k be a parameter between 1 and $\Delta = \max(\Delta_1, \Delta_2)$. Let $P_{\geq k}$ be the set of unordered pairs (x, z) such that x, z are on the same side of the bipartite graph G, and that $d_x, d_z \geq k$. Let $\# \bowtie_{\geq k}$ be the number of butterflies B such that $\exists \{x, z\} \in P_{\geq k}$, $\{x, z\} \subseteq B$. Here we abuse notation and use B to mean the vertices of a butterfly B.

In Line 11 of Algorithm 5, we construct $T_C[(x, z)]$s for pairs (x, z) with the largest $W[(x, z)]$s until the total number of butterflies in these largest $W[(x, z)]$s exceeds $(1 - \alpha)numB$. The rest of the butterflies will be maintained by T_E.

Intuitively, for proving the efficiency of `GSI`, we would like to show that a small number of $\{x, z\}$ (those in $P_{\geq k}$) covers a large fraction ($\# \bowtie_{\geq k} / numB$) of the total number of butterflies. Formally, we can prove that for some k, `GSI` has query complexity $\widetilde{O}(|P_{\geq k}|)$ and expected space complexity.

$O\left(\# \bowtie_{\geq 1} - \# \bowtie_{\geq k} + \sum_{(x,z) \in P_{\geq k}} |T_C[(x, z)]|\right)$. To show this, let's consider an algorithm similar to Algorithm 5. The modified algorithm changes the condition on Line 12 from "$num \geq \alpha \cdot numB$" to "$\{x, z\} \in P_{\geq k}$". Then T_C contains $|P_{\geq k}|$ elements (one for each $(x, z) \in P_{\geq k}$), and T_E contains $\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}$ butterflies. The time complexity of the modified algorithm is $\widetilde{O}(|P_{\geq k}|)$ and that the expected space usage of it is

$O\left(\# \bowtie_{\geq 1} - \# \bowtie_{\geq k} + \sum_{(x,z) \in P_{\geq k}} |T_C[(x, z)]|\right)$.

In the original `GSI` (Algorithm 5), for a fixed choice of k, we can choose α properly such that the condition on Line 12 evaluates to true for the first $|P_{\geq k}|$ iterations, i.e., after the first $|P_{\geq k}|$ iterations of the loop, num is no more than $\alpha \cdot numB$. The query time of `GSI` is bounded by $\widetilde{O}(|P_{\geq k}|)$. The space usage of `GSI` is bounded above by that of the modified algorithm, because the sets in W are sorted with decreasing order of sizes. Each set $W[(x, z)]$ costs $|W[(x, z)]|$ space if it is maintained in T_C and $\binom{|W[(x,z)]|}{2}$ space if it is maintained in T_E. `GSI` costs less space because it maintains larger sets in T_C, compared to the modified algorithm. These will automatically translates to the same bounds for `GSI`.

For simplicity, we use V_1, V_2 to denote U, L. We first calculate the expected number of edges, m, of G. m can be calculated by Δ_i, γ_i, n_i for either $i = 1$ or $i = 2$. For the model to be consistent, we require that the m's calculated by $i = 1$ and $i = 2$ are equal.

Lemma C.1. *For any* $i \in \{1, 2\}$, *if* $\gamma_i \in (2, 3)$,

$$m = \left(1 + O\left(\Delta_i^{2-\gamma_i}\right)\right) \frac{n_i}{s_i(\gamma_i - 2)}.$$

Proof. Let x be a fixed vertex in V_i.

$$\begin{aligned}
m =& n_i \mathbb{E}\left[\deg_x\right] && \text{(by linearity of expectation)} \\
=& n_i \mathbb{E}\left[d_x\right] \\
=& n_i \frac{1}{s_i} \sum_{i=1}^{\Delta_i} i^{1-\gamma_i} \\
=& \left(1 + O\left(\Delta_i^{2-\gamma_i}\right)\right) \frac{n_i}{s_i(\gamma_i - 2)}.
\end{aligned}$$

□

Next, we estimate the expectation of $|P_{\geq k}|$ for bounding the query time.

Lemma C.2. $\mathbb{E}\left[|P_{\geq k}|\right] = \left(1 + O\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)\left(\frac{n_1^2 s_{1,k}^2}{2s_1^2} + \frac{n_2^2 s_{2,k}^2}{2s_2^2}\right).$

Proof. Recall that for any pair of vertices x, z, $(x, z) \in P_{\geq k}$ if $x, z \in V_i$ and $d_x, d_z \geq k$. The expected number of such pairs for a fixed $i \in \{1, 2\}$ is

$$\begin{aligned}
&\binom{n_i}{2} \mathbb{P}\left[d_x \geq k\right] \mathbb{P}\left[d_z \geq k\right] \\
=&\binom{n_i}{2} \left(\frac{\sum_{d=k}^{\Delta_d} d^{-\gamma_i}}{s_i}\right)^2 \\
=&\left(1 + O\left(\frac{1}{n_i}\right)\right) \frac{n_i^2 s_{i,k}^2}{2s_i^2}.
\end{aligned}$$

Summing over $i = \{1, 2\}$ gives

$$\mathbb{E}\left[|P_{\geq k}|\right] = \left(1 + O\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)\left(\frac{n_1^2 s_{1,k}^2}{2s_1^2} + \frac{n_2^2 s_{2,k}^2}{2s_2^2}\right).$$

□

Lastly, we calculate $\mathbb{E}\left[\# \bowtie_{\geq k}\right]$, and $\mathbb{E}\left[\# \bowtie_{\geq 1}\right]$. The difference of these two numbers will bound the space usage.

Lemma C.3.

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{\geq k}\right] \\
&\geq \left(1+O\left(\Delta_1^{\gamma_1-3}+\Delta_2^{\gamma_2-3}\right)\right)\frac{n_1^2n_2^2}{4m^4} \\
&\frac{\left(\Delta_1^{3-\gamma_1}-k^{3-\gamma_1}\right)\left(\Delta_2^{3-\gamma_2}-k^{3-\gamma_2}\right)}{s_1s_2(3-\gamma_1)(3-\gamma_2)}.
\end{aligned}
$$

Proof. To get a lower bound for $\mathbb{E}\left[\# \bowtie_{\geq k}\right]$, we use the quantity $\# \bowtie_{both \geq k}$ be the number of butterflies $B = (x, y, z, w)$ such that both (x, z) and (y, w) are from $P_{\geq k}$.

$\# \bowtie_{both \geq k}$ is no more than $\# \bowtie_{\geq k}$ which is the number of butterflies such that at least one of (x, z) and (y, w) are from $P_{\geq k}$.

We sum up the probability that x, y, z, w form a butterfly for $x, z \in V_1, y, w \in V_2$.

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{both \geq k}\right] \\
=&\binom{n_1}{2}\binom{n_2}{2}\sum_{d_1,d_3\in[k,\Delta_1],d_2,d_4\in[k,\Delta_2]}\mathbb{P}\left[d_x=d_1\right]\mathbb{P}\left[d_y=d_2\right] \\
&\mathbb{P}\left[d_z=d_3\right]\mathbb{P}\left[d_w=d_4\right]\frac{d_xd_y}{m}\frac{d_xd_w}{m}\frac{d_zd_y}{m}\frac{d_zd_w}{m} \\
=&\binom{n_1}{2}\binom{n_2}{2}\sum_{d_1,d_3\in[k,\Delta_1],d_2,d_4\in[k,\Delta_2]}\mathbb{P}\left[d_x=d_1\right]\mathbb{P}\left[d_y=d_2\right] \\
&\mathbb{P}\left[d_z=d_3\right]\mathbb{P}\left[d_w=d_4\right]\frac{d_x^2d_y^2d_z^2d_w^2}{m^4} \\
=&\binom{n_1}{2}\binom{n_2}{2}\frac{1}{m^4}\left(\sum_{d_1,d_3\in[k,\Delta_1]}\mathbb{P}\left[d_x=d_1\right]\mathbb{P}\left[d_z=d_3\right]d_x^2d_z^2\right) \\
&\left(\sum_{d_2,d_4\in[k,\Delta_2]}\mathbb{P}\left[d_y=d_2\right]\mathbb{P}\left[d_w=d_4\right]d_y^2d_w^2\right) \\
=&\binom{n_1}{2}\binom{n_2}{2}\frac{1}{m^4}\left(\sum_{d_1\in[k,\Delta_1]}\mathbb{P}\left[d_x=d_1\right]d_x^2\right)^2 \\
&\left(\sum_{d_2\in[k,\Delta_2]}\mathbb{P}\left[d_y=d_2\right]d_y^2\right)^2 \\
=&\binom{n_1}{2}\binom{n_2}{2}\frac{1}{m^4}\mathbb{E}^2\left[\mathbf{1}_{d_{v_1}\geq k}d_{v_1}^2\right]\mathbb{E}^2\left[\mathbf{1}_{d_{v_2}\geq k}d_{v_2}^2\right]
\end{aligned}
$$

where v_1 (v_2) is an arbitrary vertex from V_1 (V_2). We next calculate the expectations in the equation above. For any $i \in \{1, 2\}$,

$$
\begin{aligned}
&\mathbb{E}\left[\mathbf{1}_{d_{v_i}\geq k}d_{v_i}^2\right] \\
=&\sum_{d=k}^{\Delta_i}\mathbb{P}\left[d_{v_i}=d\right]d^2 \\
=&\frac{1}{s_i}\sum_{d=k}^{\Delta}d^{2-\gamma_i} \\
=&\left(1+O\left(\Delta_i^{\gamma_i-3}\right)\right)\frac{\Delta_i^{3-\gamma_i}-k^{3-\gamma_i}}{s_i(3-\gamma_i)}.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{both \geq k}\right] \\
=&\binom{n_1}{2}\binom{n_2}{2}\frac{1}{m^4}\mathbb{E}^2\left[\mathbf{1}_{d_{v_1}\geq k}d_{v_1}^2\right]\mathbb{E}^2\left[\mathbf{1}_{d_{v_2}\geq k}d_{v_2}^2\right] \\
=&\binom{n_1}{2}\binom{n_2}{2}\frac{1}{m^4}\Pi_{i=1}^2\left(1+O\left(\Delta_i^{\gamma_i-3}\right)\right)\frac{\Delta_i^{3-\gamma_i}-k^{3-\gamma_i}}{s_i(3-\gamma_i)} \\
=&\left(1+O\left(\Delta_1^{\gamma_1-3}+\Delta_2^{\gamma_2-3}\right)\right)\frac{n_1^2n_2^2}{4m^4} \\
&\frac{\left(\Delta_1^{3-\gamma_1}-k^{3-\gamma_1}\right)\left(\Delta_2^{3-\gamma_2}-k^{3-\gamma_2}\right)}{s_1s_2(3-\gamma_1)(3-\gamma_2)}.
\end{aligned}
$$

□

Similar to Lemma C.3, we can estimate the expectation of $numB = \# \bowtie_{\geq k}$.

Lemma C.4.

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{\geq 1}\right] \\
=&\left(1+O\left(\Delta_1^{\gamma_1-3}+\Delta_2^{\gamma_2-3}\right)\right)\frac{n_1^2n_2^2}{4m^4} \\
&\frac{\Delta_1^{3-\gamma_1}\Delta_2^{3-\gamma_2}}{s_1s_2(3-\gamma_1)(3-\gamma_2)}.
\end{aligned}
$$

Proof. The proof is identical to that of Lemma C.3 with $k = 1$. We may replace the $\geq$ to $=$ because when $k = 1$, $\# \bowtie_{both \geq k}$ is equal to $\# \bowtie_{\geq k}$. □

Double-sided power-law bipartite graphs. In the double-sided power-law model, both γ_1 and γ_2 are in the range $(2, 3)$. In this case, we have that s_i is a constant for $i = 1, 2$. We also have that $m = O(n)$.

We define $n = max(n_1, n_2)$, $\Delta = max(\Delta_1, \Delta_2)$, and $\gamma = min(\gamma_1, \gamma_2)$. We may calculate

$\mathbb{E}\left[\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}\right]$ by Lemma C.4 and Lemma C.3.

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}\right] \\
=&O\left(\left(\Delta_1^{\gamma_1-3}+\Delta_2^{\gamma_2-3}\right)\frac{n_1^2n_2^2}{m^4}\left(k^{3-\gamma_1}\Delta_2^{3-\gamma_2}+k^{3-\gamma_2}\Delta_1^{3-\gamma_1}\right)\right) \\
=&O\left(\frac{n_1^2n_2^2}{m^4}\left(\left(\frac{k}{\Delta_1}\right)^{3-\gamma_1}\Delta_2^{3-\gamma_2}+\left(\frac{k}{\Delta_2}\right)^{3-\gamma_2}\Delta_1^{3-\gamma_1}+k^{3-\gamma_1}+k^{3-\gamma_2}\right)\right) \\
=&O\left(\Delta^{6-2\gamma}\right)
\end{aligned}
$$

for $k \leq \Delta$. When k is no more than the smaller of Δ_1 and Δ_2, we have a sharper bound.

$$
\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}\right] \\
=&O\left(\frac{n_1^2n_2^2}{m^4}\left(\Delta_2^{3-\gamma_2}+\Delta_1^{3-\gamma_1}+k^{3-\gamma_1}+k^{3-\gamma_2}\right)\right) \qquad (k \leq \Delta_1, k \leq \Delta_2) \\
=&O\left(\frac{n^4}{m^4}\Delta^{3-\gamma}\right) \\
=&O\left(\Delta^{3-\gamma}\right)
\end{aligned}
$$

for $k \leq \min(\Delta_1, \Delta_2)$. By Lemma C.2,

$$
\begin{aligned}
&\mathbb{E}\left[|P_{\geq k}|\right] \\
=&O\left(n^2k^{2-2\gamma}\right).
\end{aligned}
$$

Note that the two bounds above on space usage do not depend on k. Thus, we may choose the largest possible k to reduce the query time in each case.

- We may choose $k = \Delta$ so that all butterflies are maintained by T_E. The query time is $\widetilde{O}(1)$ and the expected space usage is $\mathbb{E}\left[\# \bowtie_{\geq 1}\right] = O(\Delta^{6-2\gamma})$.

- We may also choose $k = \min(\Delta_1, \Delta_2)$ so that the expected query time is $\widetilde{O}(\mathbb{E}\left[|P_{\geq k}|\right]) = \widetilde{O}(n^2 \min(\Delta_1, \Delta_2)^{2-2\gamma})$ and the expected space usage for T_E is $O(\Delta^{3-\gamma})$. Note that we need to consider the space usage of T_C. This can be bounded by

$$\begin{aligned}
&\sum_{i=1}^{2} n_i \mathbb{E}\left[\deg_{x_i}^2\right] \\
=&\sum_{i=1}^{2} n_i \frac{\sum_{d=1}^{\Delta_i} d^{2-\gamma_i}}{s_i} \\
=&O\left(\sum_{i=1}^{2} n_i \Delta_i^{3-\gamma_i}\right) \\
=&O\left(n\Delta^{3-\gamma}\right)
\end{aligned}$$

where $x_i \in V_i$ for $i = 1, 2$. The total expected space usage is $O\left(n\Delta^{3-\gamma}\right)$.

Single-sided power-law bipartite graphs. In the single-sided power-law model, we have $\gamma_1 \in (2, 3)$, $\gamma_2 = 0$, and $\Delta_1 > \Delta_2$. In this case, s_1 is a constant and $s_2 = \Delta_2$. We also have that $m = \Theta(n_1) = \Theta(n_2\Delta_2)$.

We define $n = max(n_1, n_2)$, $\Delta = max(\Delta_1, \Delta_2)$.We choose $\Delta_2 < k \leq \Delta_1$. We may calculate $\mathbb{E}\left[\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}\right]$ by Lemma C.4 and Lemma C.3.

$$\begin{aligned}
&\mathbb{E}\left[\# \bowtie_{\geq 1} - \# \bowtie_{\geq k}\right] && (1)\\
=&O\left(\left(\Delta_1^{\gamma_1 - 3} + \Delta_2^{\gamma_2 - 3}\right) \frac{n_1^2 n_2^2}{m^4} \left(k^{3-\gamma_1}\Delta_2^{3-\gamma_2} + k^{3-\gamma_2}\Delta_1^{3-\gamma_1}\right)\right) && (2)\\
=&O\left(\frac{n_1^2 n_2^2}{m^4} \left(\left(\frac{k}{\Delta_1}\right)^{3-\gamma_1} \Delta_2^{3-\gamma_2} + \left(\frac{k}{\Delta_2}\right)^{3-\gamma_2} \Delta_1^{3-\gamma_1} + k^{3-\gamma_1} + k^{3-\gamma_2}\right)\right) && (3)\\
=&O\left(\frac{n_2^2}{m^2} \left(\frac{\Delta_1^{6-\gamma_1}}{\Delta_2^3} + \Delta_2^3\right)\right) && (k \leq \Delta_1, \Delta_1 > \Delta_2)\\
=&O\left(\Delta_2 + \left(\frac{\Delta_1^{6-\gamma_1}}{\Delta_2^5}\right)\right). && (\tfrac{n_2}{m} = \Theta\left(\tfrac{1}{\Delta_2}\right))
\end{aligned}$$

Note that the bound above does not depend on k. Thus, it is an upper bound of $\mathbb{E}\left[\# \bowtie_{\geq 1}\right]$. We may set $\alpha > 1$ so that GSI maintain all butterflies in T_E. This results in an expected query time of $\widetilde{O}(1)$. □