

# Personalized Graph Summarization: Formulation, Scalable Algorithms, and Applications (Online Appendix)

## B. Proof of Lemma 1

**Implementation Details and Lemma 2.** We let  $E_{AB} := \{\{u, v\} \in E : u \in A \text{ and } v \in B\}$  be the set of edges between  $A$  and  $B$ ; and let  $W_{AB}^{(T)} := \sum_{\{u, v\} \in E_{AB}} W_{uv}^{(T)}$  be the sum of their personalized weights. We also let  $W_A^{(T)} := \sum_{u \in A} \frac{\alpha - D(u, T)}{\sqrt{Z}}$  and  $\bar{W}_A^{(T)} := \sum_{u \in A} \frac{\alpha - 2 \times D(u, T)}{Z}$ . PEGASUS maintains the following values up-to-date:

$$W_A^{(T)} \text{ and } \bar{W}_A^{(T)}, \quad \forall A \in S \text{ and} \quad (13)$$

$$W_{AB}^{(T)}, \quad \forall \{A, B\} \in \Pi_S \text{ where } E_{AB} \neq \emptyset. \quad (14)$$

To this end, PEGASUS initializes them by BFS in  $O(|V| + |E|)$  time and updates them for each added/removed supernode and superedge, as described below. By the definitions of  $W_A^{(T)}$ ,  $\bar{W}_A^{(T)}$  and  $W_{AB}^{(T)}$ , the size of values in Eq. (13) is  $O(|V|)$ , and the size of those in Eq. (14) is  $O(|E|)$ .

**Lemma 2.** For any supernodes  $A, B \in S$  in a summary graph  $\bar{G}$ ,  $Cost_{AB}^{(T)}(\bar{G})$  can be computed in  $O(1)$  time.

*Proof.* It suffices to show that computing  $RE_{AB}^{(T)}(\bar{G})$  takes  $O(1)$  time since computing the remaining terms trivially takes  $O(1)$  time. Eq. (6) and the definitions above imply that

$$RE_{AB}^{(T)}(\bar{G}) = \begin{cases} W_A^{(T)} W_B^{(T)} - W_{AB}^{(T)} - \bar{W}_A^{(T)}, & \text{if } \{A, B\} \in P \text{ and } A = B, \\ W_A^{(T)} W_B^{(T)} - W_{AB}^{(T)}, & \text{if } \{A, B\} \in P \text{ and } A \neq B, \\ W_{AB}^{(T)}, & \text{otherwise.} \end{cases} \quad (15)$$

All terms in Eq. (15) are maintained as described above, and thus  $RE_{AB}^{(T)}(\bar{G})$  can be computed in  $O(1)$  time.  $\square$

**Proof of Lemma 1.** For the update of  $\bar{G}$ , lines 6-10 of Alg. 2 are executed, and lines 6-7 trivially take  $O(1)$  time. Lines 8-9 take  $O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$  time, as shown below.

Let  $E_{XY} := \{\{u, v\} \in E : u \in X \text{ and } v \in Y\}$  be the set of edges between supernodes  $X \in S$  and  $Y \in S$ . According to our initialization and update schemes,  $\{X, Y\} \notin P$  holds for any supernodes  $X, Y \in S$  where  $E_{XY} = \emptyset$ . This is because adding such  $\{X, Y\}$  to  $P$  increases  $Size(\bar{G})$ ,  $RE^{(T)}(\bar{G})$ , and thus  $Cost^{(T)}(\bar{G})$ . Hence, for any supernode  $X \in S$ , the number of superedges incident to  $X$  is  $O(\sum_{x \in X} |N_x|)$ .

**(i) Updating intermediate results:** As the first step, the intermediate results in Eq. (13) and (14) are updated as follows:

$$\begin{aligned} W_{(A \cup B)(A \cup B)}^{(T)} &\leftarrow W_{AA}^{(T)} + W_{BB}^{(T)} + W_{AB}^{(T)}, \\ W_{(A \cup B)X}^{(T)} &\leftarrow W_{AX}^{(T)} + W_{BX}^{(T)}, \forall X \in S \text{ where } E_{(A \cup B)X} \neq \emptyset, \\ W_{A \cup B}^{(T)} &\leftarrow W_A^{(T)} + W_B^{(T)}, \text{ and } \bar{W}_{A \cup B}^{(T)} \leftarrow \bar{W}_A^{(T)} + \bar{W}_B^{(T)} \end{aligned}$$

As discussed above, the number of  $X \in S$  where  $E_{(A \cup B)X} \neq \emptyset$  is  $O(\sum_{u \in A \cup B} |N_u|) = O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$ , and each update takes  $O(1)$  time. Thus, all updates take  $O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$  time.

**(ii) Line 8:** For line 8, removing the superedges incident to  $A$ , whose number is  $O(\sum_{u \in A} |N_u|)$ , and removing the superedges incident to  $B$ , whose number is  $O(\sum_{v \in B} |N_v|)$ , takes  $O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$  time.

**(iii) Line 9:** PEGASUS minimizes  $Cost_{A \cup B}^{(T)}(\bar{G})$  by adding each superedge  $\{(A \cup B), X\}$  to  $P$  if and only if it decreases  $Cost_{(A \cup B)X}^{(T)}(\bar{G})$ . As described above, the number of potential superedges incident to  $A \cup B$  is  $O(\sum_{u \in A \cup B} |N_u|) = O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$ , and by Lemma 2, computing  $Cost_{(A \cup B)X}^{(T)}(\bar{G})$  takes  $O(1)$  time. Hence, line 9 takes  $O(\sum_{u \in A} |N_u| + \sum_{v \in B} |N_v|)$  time.  $\square$

## C. The Effects of Relative Cost Reduction Function

We compared PEGASUS, which uses Eq. (11), and a variant that uses Eq. (10) instead, in order to demonstrate the effect of using the relative cost reduction in Eq. (11). As shown in Fig. 13, using Eq. (11) led to better summary graphs, where queries can be answered more accurately, than using Eq. (10). See the last paragraph of Sect. III-B for further discussion.

## D. Results of PHP Queries (Q3, Q5)

We report the accuracy of answers to PHP queries that is omitted in Sect. V-D and Sect. V-F.

As shown in Fig. 14, PEGASUS outperformed the state-of-the-art non-personalized graph summarization algorithms. For example, for the Amazon0601 dataset, the answers to PHP queries were up to  $3.86 \times$  and  $1.24 \times$  more accurate in terms of SMAPE and SC, respectively.

As shown in Fig. 15, the application of PEGASUS was more helpful for communication-free distributed multi-query processing than the considered graph partitioning algorithms. For example, for the Caida dataset, the answers to PHP queries were up to  $3.22 \times$  and  $1.34 \times$  more accurate in terms of SMAPE and SC, respectively.

## E. The Effects of Sampling Methods for Target Node Set

Recall that, in Sect. V of the main paper, we chose target nodes  $T$  uniformly at random. In order to show that PEGASUS is not specialized to this way of choosing target nodes, we performed experiments in Sect. V of the main paper using target node sets chosen in a different manner. Specifically, we chose a node  $u$  uniformly at random and sampled  $|T|$  adjacent target nodes by breadth first search (BFS) from  $u$  on the input graph. All other experimental settings were the same.

The results are consistent with those in Sect. V of the main paper. Specifically, Fig. 16 shows that PEGASUS successfully personalized summary graphs. In Figs. 17-18, PEGASUS was one of the most scalable algorithms, and it showed linear scalability. How (a) the degree of personalization  $\alpha$  and (b) the parameter for adaptive thresholding  $\beta$  affected query accuracy in Figs. 19-20 is consistent with the results in the main paper. In Fig. 21, PEGASUS significantly outperformed all other non-personalized algorithms.

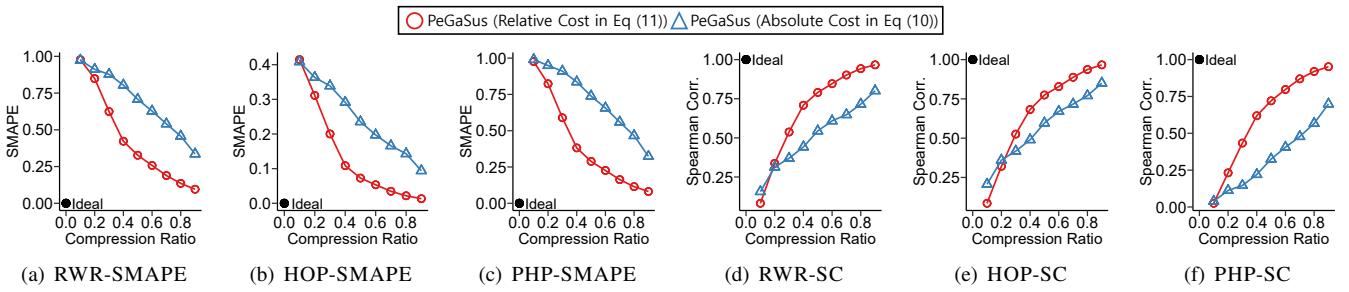


Fig. 13. [Using Eq. \(11\) for greedy search is effective](#). Queries are answered more accurately when Eq. (11) is used than when Eq. (10) is used.

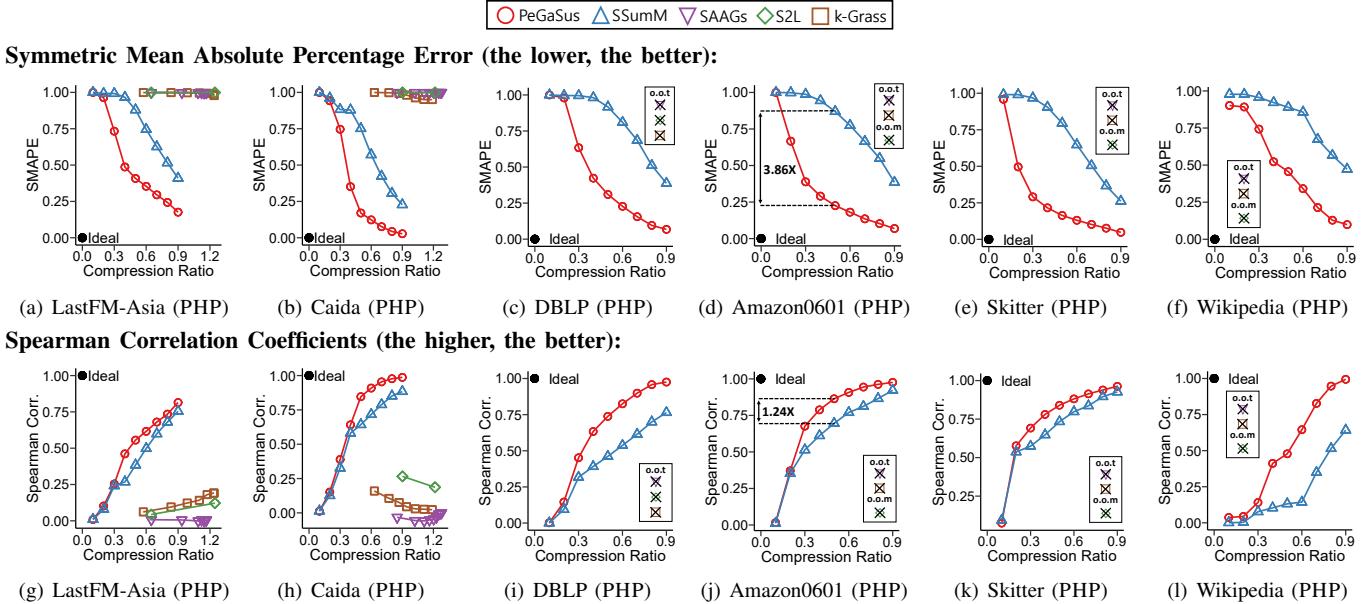


Fig. 14. [PEGASUS gives summary graphs where queries on target nodes randomly sampled are answered most accurately](#). o.o.t: out of time (> 48hours). o.o.m: out of memory (> 128GB). The degree of personalization  $\alpha$  is fixed to 1.25, and the size of the target node set is fixed to 100.

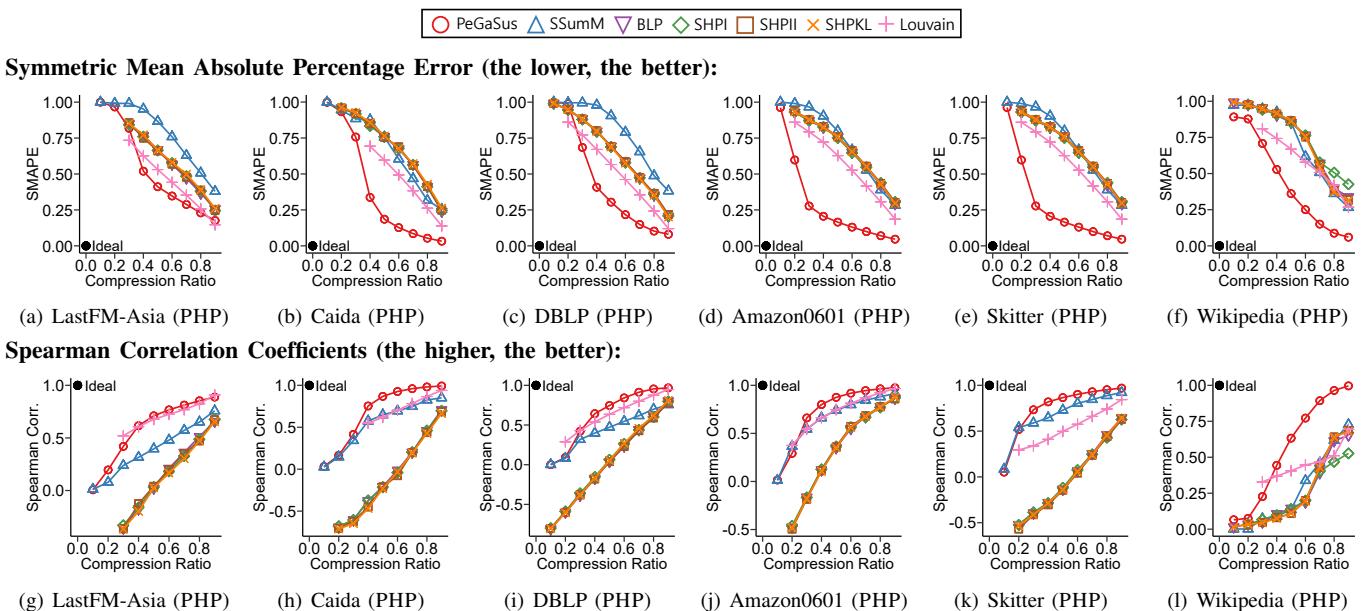


Fig. 15. [PEGASUS is useful for “communication-free” distributed multi-query processing](#). Queries are answered more accurately from distributed personalized summary graphs (PEGASUS) than from non-personalized summary graphs (SSUMM) or distributed subgraphs (the others). The degree of personalization  $\alpha$  is fixed to 1.25. In some datasets, compression rates cannot be lowered further due to imbalance among graph partitions.

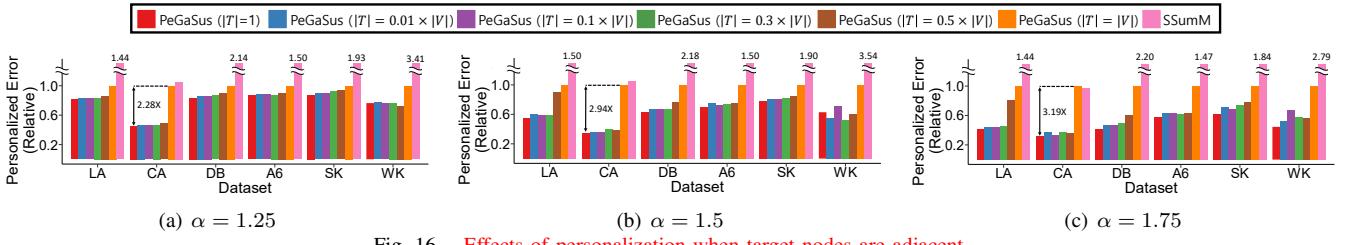


Fig. 16. Effects of personalization when target nodes are adjacent.

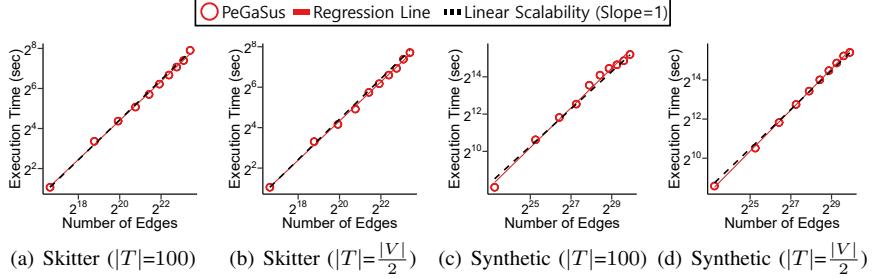


Fig. 17. Scalability of PEGASUS when target nodes are adjacent.

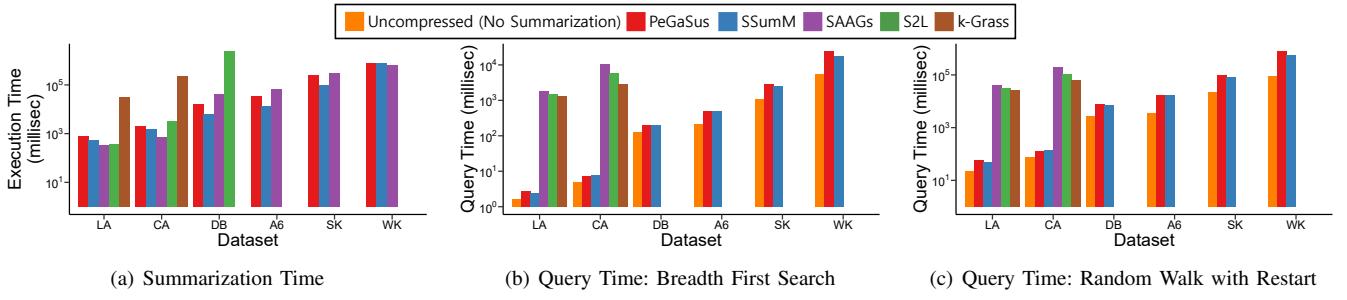


Fig. 18. Speed of summarization and query answering when target nodes are adjacent.

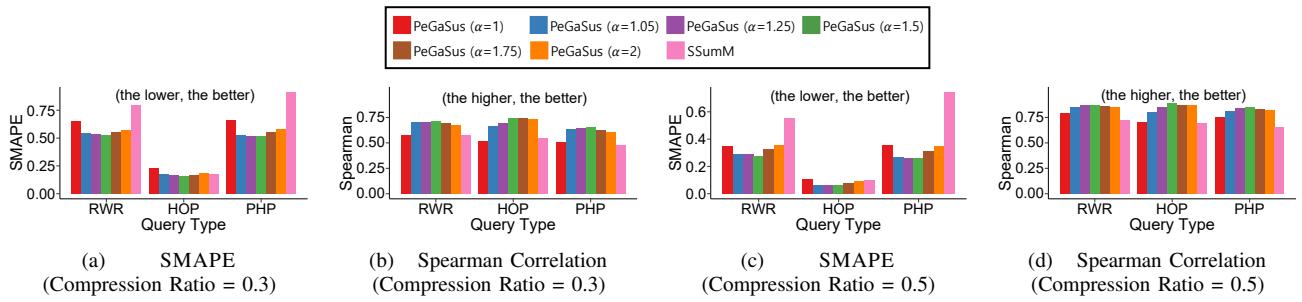


Fig. 19. Effects of the degree of personalization  $\alpha$  on the accuracy of query answering when target nodes are adjacent.

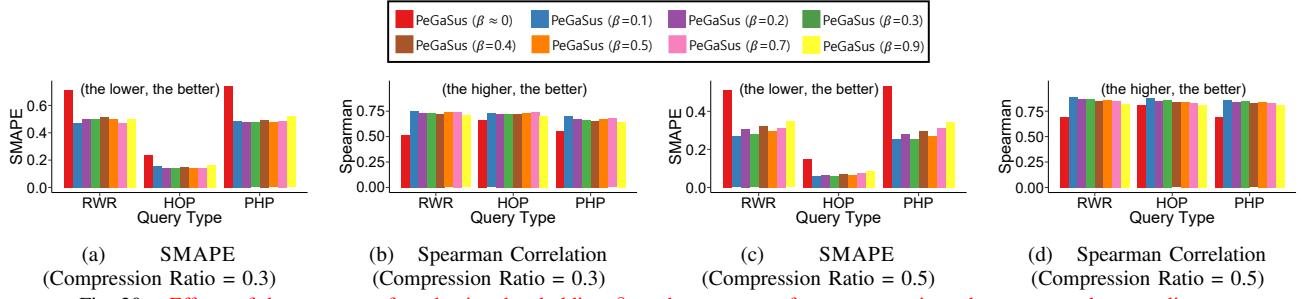
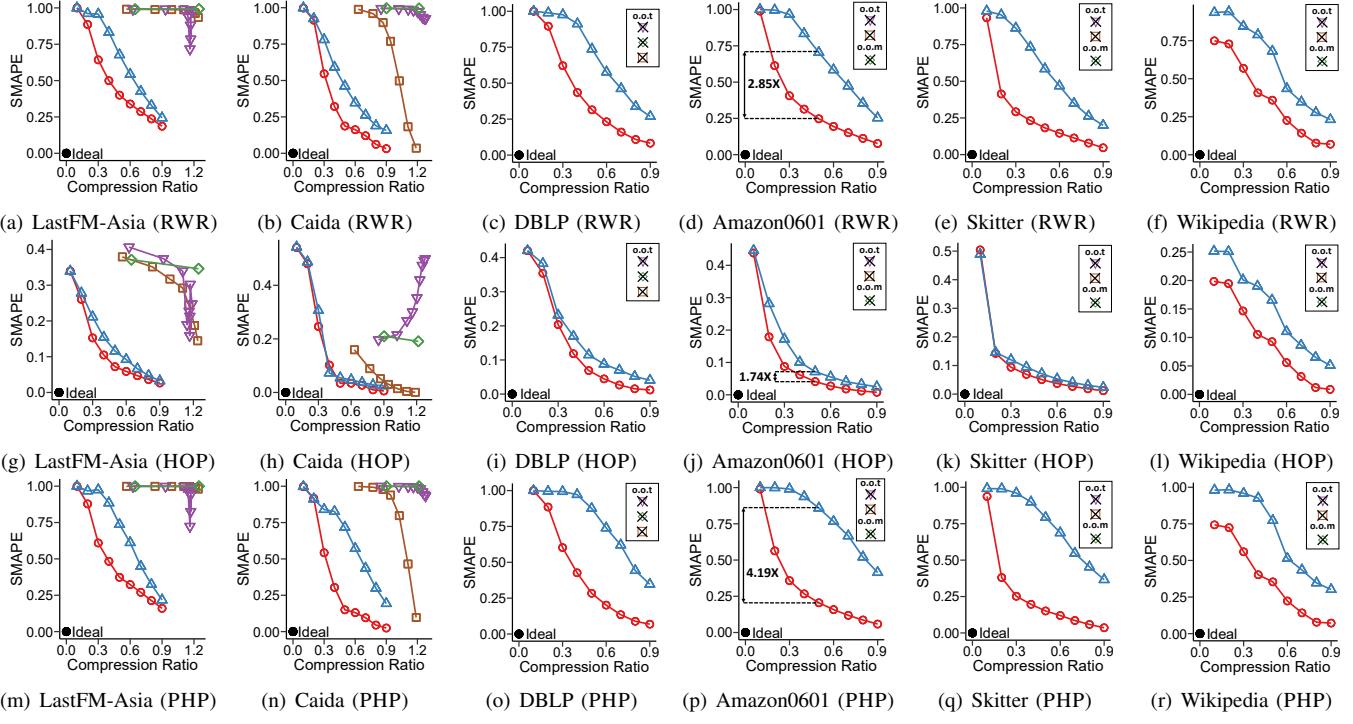


Fig. 20. Effects of the parameter for adaptive thresholding  $\beta$  on the accuracy of query answering when target nodes are adjacent.



### Symmetric Mean Absolute Percentage Error (the lower, the better):



### Spearman Correlation Coefficients (the higher, the better):

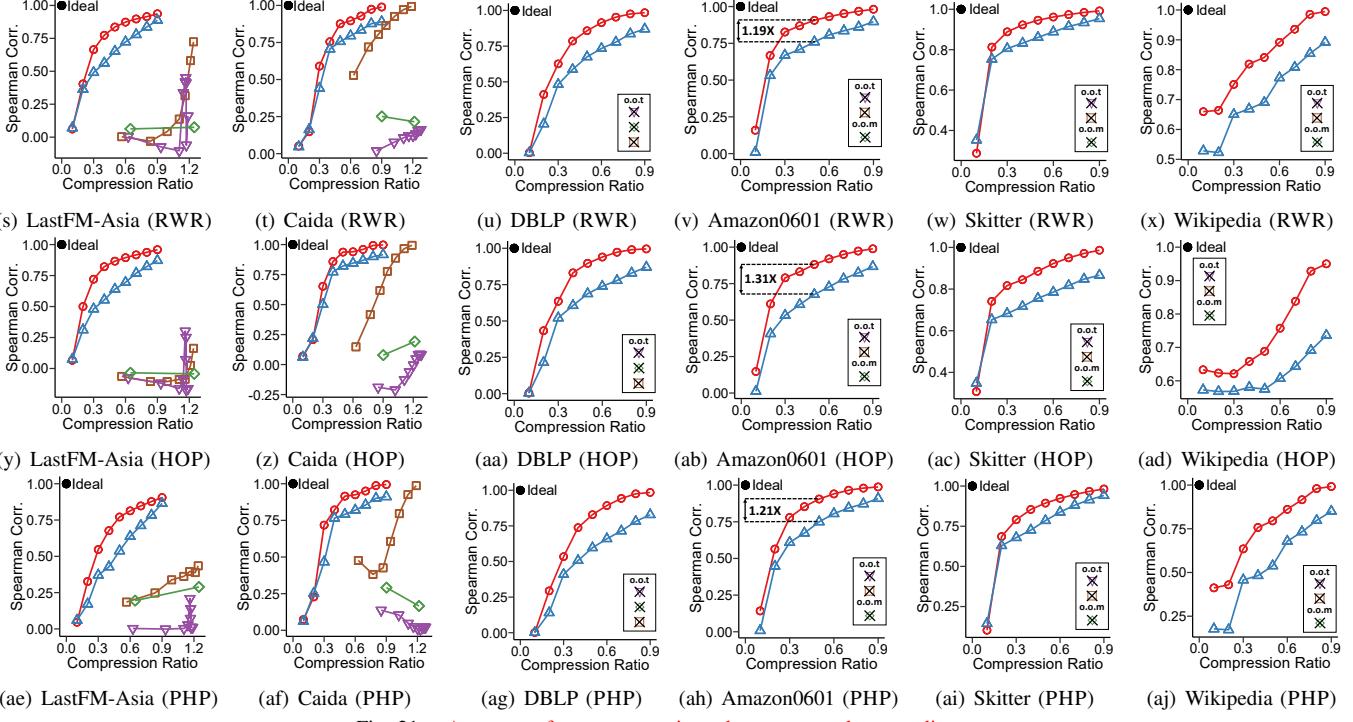


Fig. 21. Accuracy of query answering when target nodes are adjacent.