Appendix

A DATA GENERATOR EFFECTIVENESS

To illustrate the effectiveness of our generator, we follow the previous evaluation methodology [1], i.e., generate some connected communities over the social network with 150K persons and present the distribution of dataset communities statistics. We continue to use six major statistics shown in Table 1, where C is the community, $|C|$ is the number of vertices in C, $t(C)$ is the triangle number of C, $d(u, C)$ and $d(u, G)$ is the number of neighbors in C and G, respectively. $diameter(C)$ is the community's diameter, which is the maximum distance between two vertices $bridges(C)$ is the number of bridges, where the bridge is an edge whose deletion disconnects the community.

Table 1: Six major statistics and their definitions

Statistics	Definitions
Clustering Coefficient	$\frac{3 \cdot t(C)}{\sum_{u \in C}[d(u,C) \cdot (d(u,C)-1)/2]}$
Triangle Participation Ratio	$\frac{\lvert\{u \in C \mid t(u,C) > 0\}\rvert}{\lvert C \rvert}$
Bridge Ratio	$\frac{2 \cdot bridges(C)}{\sum_{u \in C} d(u,C)}$
Diameter	$\frac{diameter(C)}{\log_{10}(\lvert C \rvert) + 1}$
Conductance	$\frac{\sum_{u \in C}(d(u,G) - d(u,C))}{\sum_{u \in C} d(u,G)}$
Size(log10)	$\frac{1}{\log_{10}(\lvert C \rvert)}$

For the ground truth, we use the LiveJournal dataset and communities [2], computing the above statistics and display distributions in Figure 1. Then we generate communities over our synthetic datasets and LDBC's datasets, presenting the distributions in Figure 2 and Figure 3, respectively. These distributions exhibit high similarity and low divergence. Specifically, we use Hensen-Shannon Divergence to quantify the distribution divergence and the detailed value is listed in Table 2. Our datasets achieve averagely 2x lower divergence, verifying that our generator can generate social networks that follow the real-world distribution and are similar to real-world datasets.

Table 2: Jensen-Shannon Divergence between LiveJournal and our/LDBC's datasets

Datasets	CC	TPR	BR	Diam	Cond	Size
Ours	0.108	0.057	0.057	0.053	0.054	0.069
LDBC	0.201	0.097	0.087	0.236	0.183	0.133

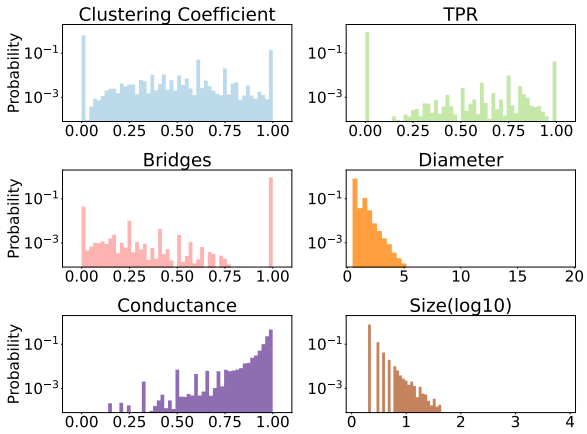


Figure 1: Statistic distribution of LiveJournal

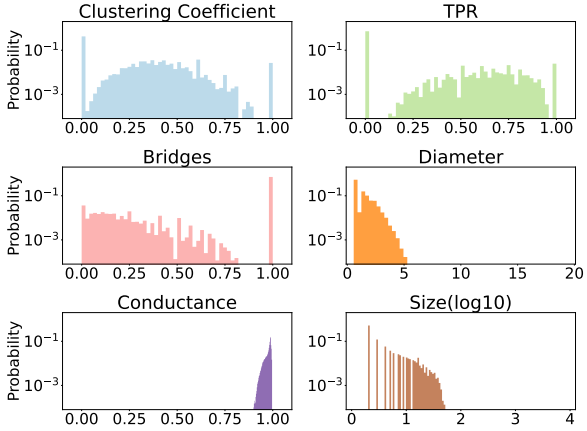


Figure 2: Statistic distribution of our synthetic datasets

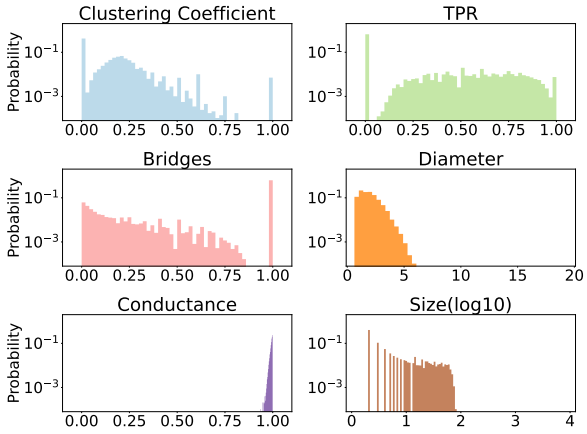


Figure 3: Statistic distribution of LDBC datasets

Table 3: Scaling Factor: the best performance over single-thread performance

Algo.	Dataset	GX	PG	FL	GR	PP	LI	GT
PR	S8-Std	3.8	5.1	8.2	25.3	29.3	32.2	—
	S8-Dense	3.8	7.8	4.5	25.2	22.6	34.9	—
	S8-Diam	3.6	2.2	8.2	24.2	18.9	32.0	—
SSSP	S8-Std	6.9	5.0	9.3	23.5	14.7	17.8	—
	S8-Dense	7.8	5.8	8.5	19.7	16.4	18.2	—
	S8-Diam	6.7	0.9	10.2	13.2	15.8	22.4	—
TC	S8-Std	—	10.7	18.7	37.2	—	21.3	19.7
	S8-Dense	—	10.5	22.2	27.5	—	22.4	30.1
	S8-Diam	—	9.4	18.1	29.6	—	23.5	20.0

Table 4: Scaling Factor: the best performance over single-machine performance

Algo.	Dataset	GX	PG	FL	GR	PP	GT
PR	S8-Std	1.7	1.6	0.8	2.3	2.5	—
	S8-Dense	1.4	1.6	0.9	2.7	4.7	—
	S8-Diam	1.6	1.5	0.9	15.6	4.1	—
SSSP	S8-Std	1.9	1.4	1.1	1.3	1.7	—
	S8-Dense	1.5	1.1	1.1	2.1	2.3	—
	S8-Diam	1.3	0.9	1.5	1.9	1.9	—
TC	S8-Std	—	2.0	2.5	2.0	—	3.1
	S8-Dense	—	3.3	2.3	3.2	—	2.7
	S8-Diam	—	2.4	2.4	3.7	—	3.1

Table 5: Scaling Factor: the best performance over single-machine performance

Algo.	Dataset	GX	PG	FL	GR	PP	GT
PR	S9-Std	3.2	2.3	0.8	5.8	5.2	—
	S9-Dense	3.8	2.2	1.0	11.5	6.1	—
	S9-Diam	3.0	2.4	0.8	6.1	5.4	—
SSSP	S9-Std	1.8	2.6	1.2	1.7	3.2	—
	S9-Dense	2.2	2.9	1.3	3.3	5.0	—
	S9-Diam	—	1.4	2.0	0.5	3.9	—
TC	S9-Std	—	—	3.3	3.2	—	3.1
	S9-Dense	—	—	2.6	4.1	—	2.7
	S9-Diam	—	—	4.7	3.9	—	3.1

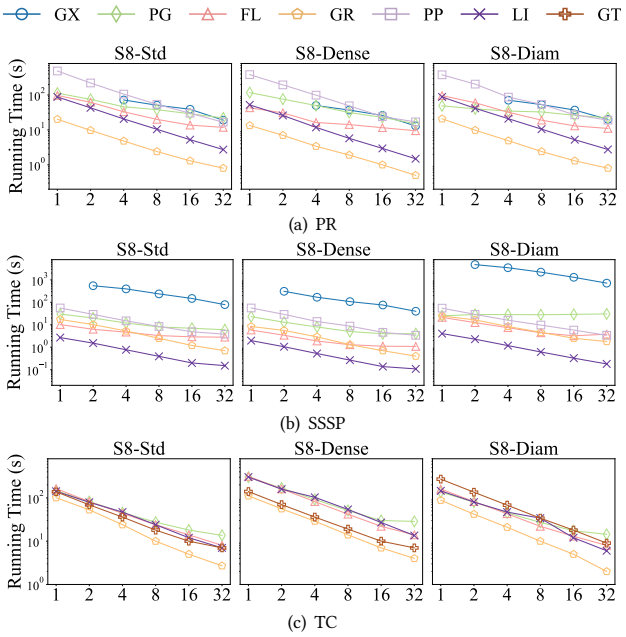


Figure 4: Running time of PR, SSSP and TC, with three datasets (Scale = 8), varying #threads

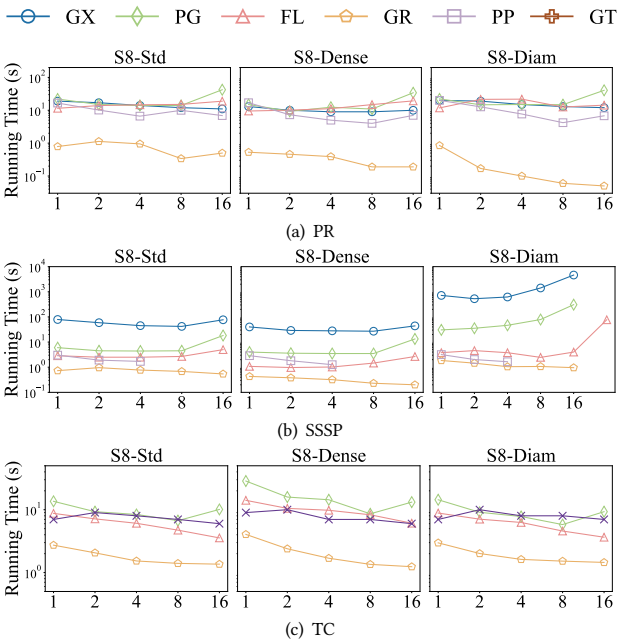


Figure 5: Running time of PR, SSSP and TC, with three datasets (Scale = 8), varying #machines

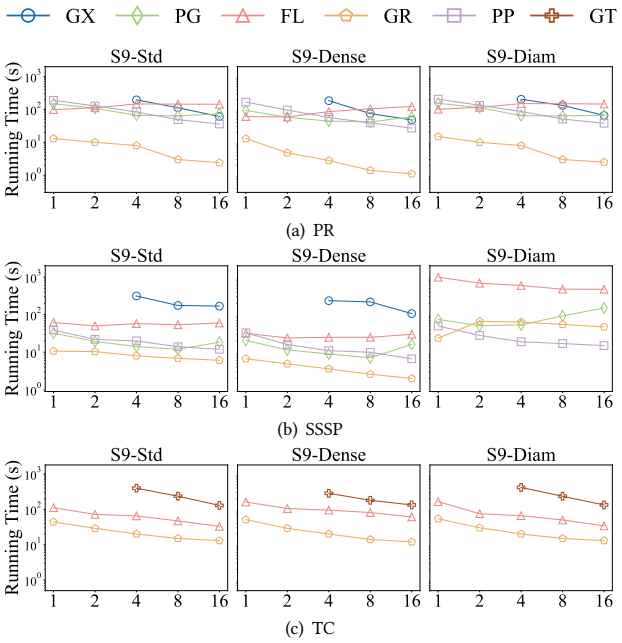
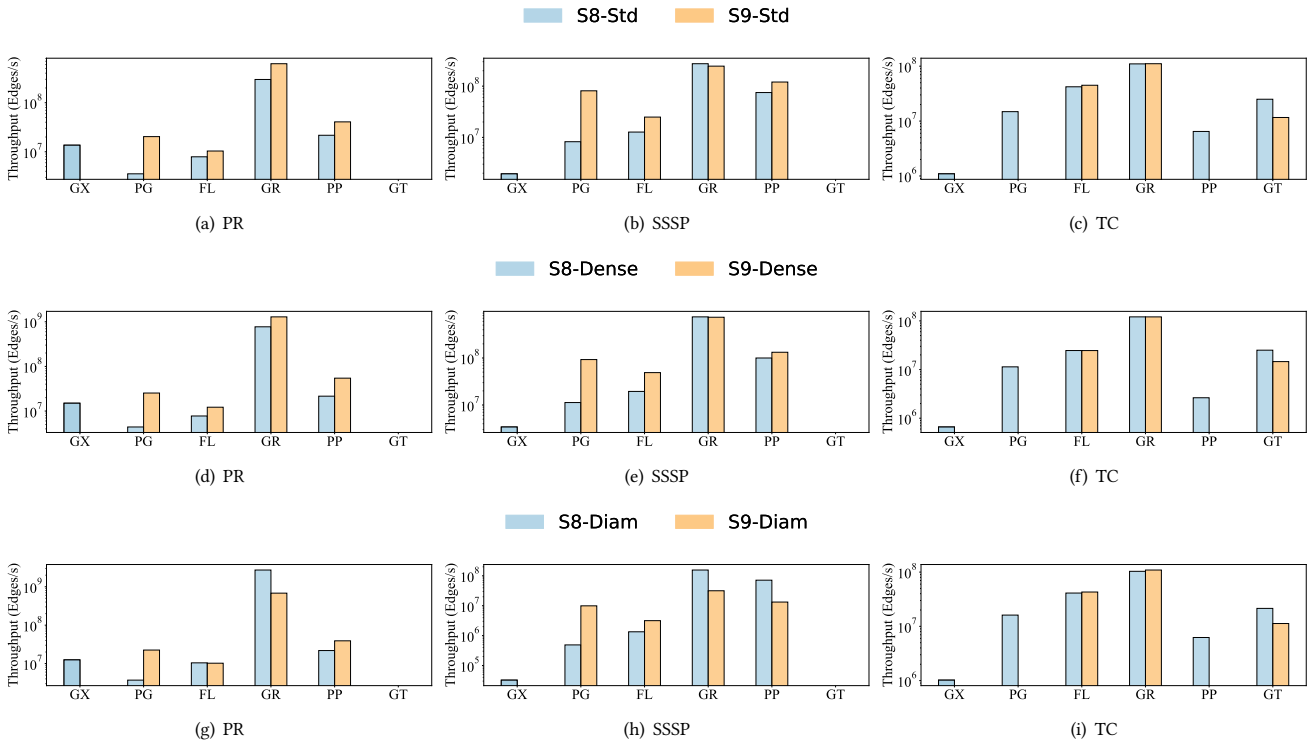


Figure 6: Running time of PR, SSSP and TC, with three datasets (Scale = 9), varying #machines



(a) PR (b) SSSP (c) TC

(d) PR (e) SSSP (f) TC

(g) PR (h) SSSP (i) TC

Figure 7: Throughput of PR, SSSP and TC on various platforms using Std datasets

B SCALABILITY

Figure 4 shows the running time of PR, SSSP, and TC on three S8 datasets (S8-Std, S8-Dense, and S8-Diam) with increasing number of threads. Figure 5 and Figure 6 presents the running time of PR, SSSP, and TC on three S9 datasets (S9-Std, S9-Dense, and S9-Diam) with increasing number of machines, respectively.

Generally, the scalability is well in Figure 4 (up to 30x speedup) but terrible in Figure 5 (even becoming worse), since computing resource is gradually approaching the upper limit of parallelism. We then enlarge the datasets with Scale = 9 and Figure 6 shows significant improvement in scalability.

The scalability is diverse with different algorithms. The quantified performance improvement is shown in Table 3, Table 4, Table 5, respectively. Among all three algorithms, TC achieves the highest scalability, since the workload is highly local and independent. However, the scalability is limited when solving PR and SSSP, which involve multiple iterations.

The scalability also varies in different datasets. The scalability is better on S8-Dense and S9-Dense, but worse on S8-Diam and S9-Diam. The major reason is that dense datasets have a more concentrated computation workload, while a dataset with the large diameter has too much computing dependency and requires more synchronizations.

C THROUGHPUT

Figure 7 presents the comparative analysis of throughput (measured in edges per second) for PageRank (PR), SSSP and Triangle Counting (TC) on various platforms using Std datasets (other datasets, Dense and Diam, show consistent results). Across all dataset types, Grape and Pregel+ consistently outperform the other platforms, showing significantly higher throughput. Flash and Grape demonstrate the best stability when handling datasets of different scales (S8 and S9), especially in the Triangle Counting algorithm. It indicates that their architecture is well-suited for varying data sizes, providing consistent performance. In contrast, PowerGraph and Pregel+ show significant improvements in performance when handling larger datasets compared to smaller ones. It suggests these platforms have well scalable architectures that leverage the increased data volume to optimize processing.

D STRESS TEST

The stress test results, shown in Table 6, reveal distinct performance variations among the graph analytics platforms when running PR on graphs of different scales. PowerGraph handles graphs up to the S9 scale, but its inefficient data loading poses challenges when processing large datasets. GraphX and Flash successfully process graphs up to the S9.5 scale. GraphX has an extreme memory consumption when executing algorithms, while Flash requires single-machine preprocessing of data. Both factors lead to high memory demands when handling large datasets. In contrast, Grape and Pregel+ successfully process all scales from S8 to S10, demonstrating advanced optimization and superior resource management.

Table 6: Stress test under varying graph scales (PR)

Platforms	S8-Std	S9-Std	S9.5-Std	S10-Std
GX	✓	✓	✓	
PG	✓	✓		
FL	✓	✓	✓	
GR	✓	✓	✓	✓
PP	✓	✓	✓	✓
GT	✓	✓	✓	✓

REFERENCES

[1] Arnau Prat-Pérez and David Domínguez-Sal. 2014. How community-like is the structure of synthetically generated graphs?. In *Second International Workshop on Graph Data Management Experiences and Systems, GRADES 2014, co-loated with SIGMOD/PODS 2014, Snowbird, Utah, USA, June 22, 2014*, Peter A. Boncz and Josep Lluís Larriba-Pey (Eds.). CWI/ACM, 7:1–7:9. https://doi.org/10.1145/2621934.2621942

[2] Jaewon Yang and Jure Leskovec. 2012. Defining and Evaluating Network Communities Based on Ground-Truth. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu (Eds.). IEEE Computer Society, 745–754. https://doi.org/10.1109/ICDM.2012.138