

SNAP-tFDP: Massively Scalable Graph Layouts via Sparse Negative Sampling

– Supplementary Material –

Xin Chen , Shuowei Hou , Yifan Wang , Mingliang Xue , Zezheng Feng ,
Oliver Deussen , Weidong Huang , and Yunhai Wang 

This supplemental material presents four sets of additional results for the submitted paper: (i) a detailed discussion of the proposed degree-weighted repulsive force; (ii) an empirical comparison of conflict rates; (iii) complete experimental results; and (iv) visualizations produced by all twelve graph layout methods on all datasets.

1 DISCUSSION OF DEGREE-WEIGHTED REPULSIVE FORCE

This section presents an ablation study together with a comparison of alternative degree-normalization schemes to examine the factors contributing to the improved cluster separation achieved by SNAP-tFDP.

Ablation Study. To comprehensively evaluate the components of SNAP-tFDP, we designed seven settings for the ablation study, as summarized in Table 1. Settings S1–S3 use the power-function-based forces of the FR model [3], whereas S4–S7 adopt the t -forces of t -FDP [17]. This design enables a direct comparison of the two force models under the same weighting strategy used in the submitted paper. Degree Weighting uses $d_i + d_j$, while Degree Normalization further divides the weight by $2|V|$, which is the final formulation adopted by SNAP-tFDP. The last setting (S7) corresponds to the complete SNAP-tFDP implementation and is included to evaluate the effect of replacing the explicit degree weight with the implicit weighting induced by negative sampling.

Table 1: Seven studied settings of the ablation study.

Setting	Power-function-based Force	t -Force	Degree Weighting	Degree Normalization	Negative Sampling
S1	✓				
S2	✓		✓		
S3	✓		✓		
S4		✓		✓	
S5		✓	✓		
S6		✓	✓	✓	
S7		✓	✓	✓	✓

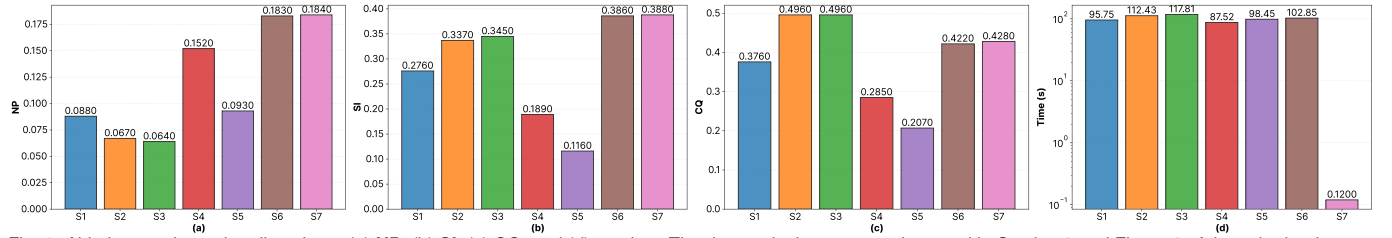


Fig. 1: Ablation results under all settings: (a) NP, (b) SI, (c) CQ, and (d) runtime. The dataset is the same as that used in Section 3 and Figure 2 of the submitted paper.

Figure 1 reports the corresponding metric results under all settings. Peak memory consumption is omitted because none of the settings requires auxiliary data structures, resulting in nearly identical memory usage. Four observations can be drawn.

- X. Chen, S. Hou and Y. Wang are with Renmin University of China. E-mail: {chenxin19961029, 2025104058, wang.yh}@ruc.edu.cn.
- Y. Wang is with Shandong University. E-mail: yfwang2001@gmail.com.
- M. Xue is with Shandong Provincial Institute of Educational Sciences. E-mail: xml95007@gmail.com.
- Z. Feng is with Northeastern University. E-mail: fengzezheng@swc.neu.edu.cn.
- O. Deussen is with University of Konstanz. E-mail: oliver.deussen@uni-konstanz.de.
- W. Huang is with University of Technology Sydney. E-mail: weidong.huang@uts.edu.au.
- Yunhai Wang is the corresponding author.

First, without degree weighting, power-function-based forces perform substantially worse than t -forces in Neighborhood Preservation [15] (NP), but worse in Silhouette Index [14] (SI) and Clustering Quality [8] (CQ). This indicates that t -forces are more effective at preserving lgraph neighborhoods, which naturally yields clusters with less overlap. However, the blurred boundaries caused by insufficient inter-cluster separation may incorrectly assign boundary nodes whose labels differ from those of the surrounding cluster, leading to low SI and CQ scores.

Second, applying only Degree Weighting consistently reduces NP. For power-function-based forces, however, it improves both SI and CQ, whereas for t -forces it degrades all three metrics. The reason is that degree weighting further amplifies the disruption of graph neighborhoods caused by power-function-based repulsion, as analyzed in t -FDP [17]. At the same time, it increases cluster separation by attracting low-degree vertices toward connected high-degree centers while strengthening the repulsion between different communities, following the same intuition as LinLog and FA2. For t -forces, however, as discussed in Section 3.2 of the submitted paper, amplifying repulsion unnecessarily enlarges equilibrium distances, resulting in looser clusters and more ambiguous boundaries, thereby degrading all three metrics (see Figure 2).

Third, adding Degree Normalization has little effect on power-function-based forces but substantially improves SI and CQ, while slightly improving NP, for t -forces. For any connected component, $d_i + d_j > 1$, whereas $(d_i + d_j)/(2|V|) < 1$, meaning that the former increases the relative strength of repulsion, while the latter reduces it. For power-function-based forces, this difference has only a limited effect because the unbounded short-range repulsion dominates the force balance, while the relative node degrees continue to promote cluster formation as discussed above. In contrast, the relative strength of repulsion is critical for t -forces. Normalized weights smaller than one move the equilibrium distance closer to zero, directly promoting compact clusters and yielding higher SI and CQ than the unweighted setting (S4). The slight improvement in NP suggests that stronger cluster contraction also helps preserve neighborhood relationships at the cluster level, consistent with our design objective.

Finally, SNAP- t FDP (S7), which replaces explicit weighting with negative sampling, achieves essentially identical visual quality to S6 on all three metrics while substantially reducing runtime. This demonstrates that the proposed edge-centric negative sampling with SGD-based optimization can implicitly realize the linear normalized weight without disturbing cluster visibility, while reducing the time complexity from quadratic, $O(|V|^2)$, to linear, $O(|E|)$.

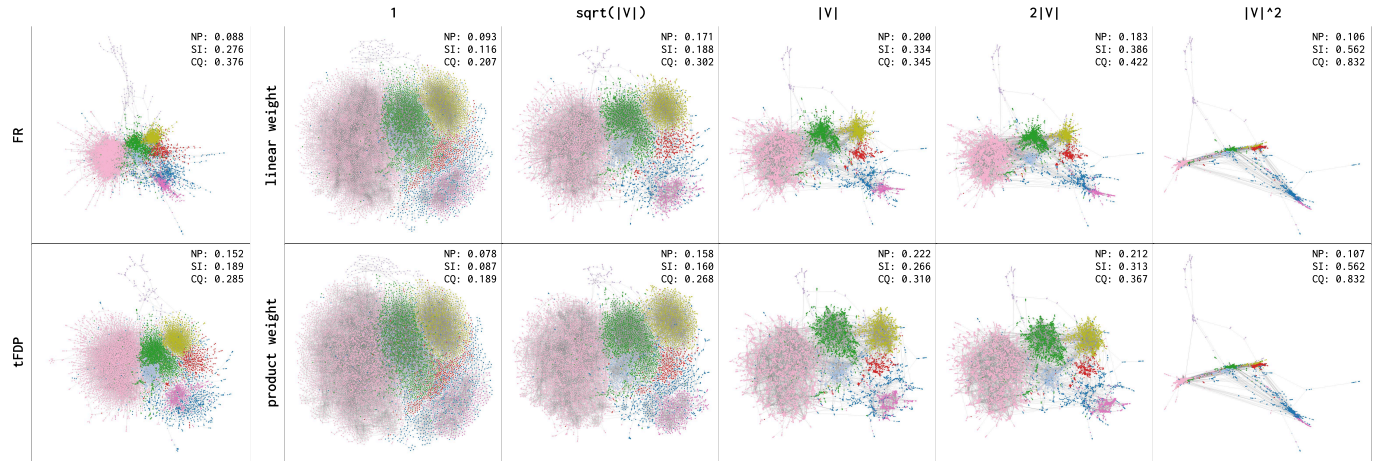


Fig. 2: Comparison of alternative degree-weighting and normalization schemes. (Left) Results of the unweighted FR and t -FDP models. (Right) Layouts obtained using linear weighting ($d_i + d_j$; upper row) and product-based weighting ($(d_i + 1)(d_j + 1)$; lower row) under five normalization schemes. The corresponding metric scores are shown in the upper-right corner of each layout. The dataset is the same as that used in Section 3 and Figure 2 of the submitted paper.

Comparison of Alternative Degree-Normalization Schemes. To better understand how degree normalization affects cluster-separation behavior, we compare five normalization schemes under both linear weighting ($d_i + d_j$) and product-based weighting ($(d_i + 1)(d_j + 1)$):

- 1 : no normalization;
- $\sqrt{|V|}$: sublinear normalization;
- $|V|$: an alternative linear normalization;
- $2|V|$: the linear normalization adopted in the submitted paper;
- $|V|^2$: quadratic normalization.

Figure 2 presents the resulting layouts and quality metrics, together with the unweighted FR and t -FDP layouts for reference. From the perspective of normalization, removing normalization further increases the relative strength of repulsion, leading to cluster expansion, blurred boundaries, and consistently worse metric values than the unweighted t -FDP model. Sublinear normalization slightly alleviates this effect but still provides insufficient cluster separation, resulting in performance comparable to unweighted t -FDP. Both linear normalization schemes outperform unweighted t -FDP on all three metrics. The alternative linear normalization ($|V|$) achieves slightly higher NP, whereas our normalization ($2|V|$) yields higher SI and CQ; however, their overall performance is of the same order. In contrast, quadratic normalization slightly improves NP while substantially increasing SI and CQ compared to unweighted t -FDP. This apparent improvement is misleading, as the excessive suppression of repulsion causes clusters to

collapse into dense singularities, obscuring both the topology and the internal cluster structure. These results indicate that linear normalization provides the best balance: it suppresses excessive repulsion at the group level while preventing cluster collapse and preserving internal structure.

From the perspective of the weighting function, product-based weighting consistently produces lower **SI** and **CQ** scores than linear weighting because it further amplifies the relative strength of repulsion. The only exception is quadratic normalization, where both weighting schemes suppress repulsion to nearly zero, producing collapsed clusters with nearly identical metric values. For **NP**, linear weighting performs better under no normalization and sublinear normalization, whereas product-based weighting achieves slightly higher scores under the two linear normalization schemes. Under quadratic normalization, the two weighting schemes produce nearly identical results. These observations suggest that when the weighting factor is greater than one, increasing repulsion progressively damages graph neighborhoods. Conversely, when the weighting factor is less than one, stronger suppression of repulsion improves cluster separation but also increasingly weakens neighborhood preservation. Since the primary objective of this work is to improve cluster visibility, linear weighting is preferable. For example, the blue and purple clusters in the lower-right corner have noticeably clearer boundaries under linear weighting.

Overall, the combination of linear weighting and linear normalization provides the best balance between cluster separation and neighborhood preservation. Moreover, the parameter k in SNAP-tFDP provides an additional mechanism for adjusting the effective strength of linear normalization. These results further support that SNAP-tFDP achieves the intended design objectives.

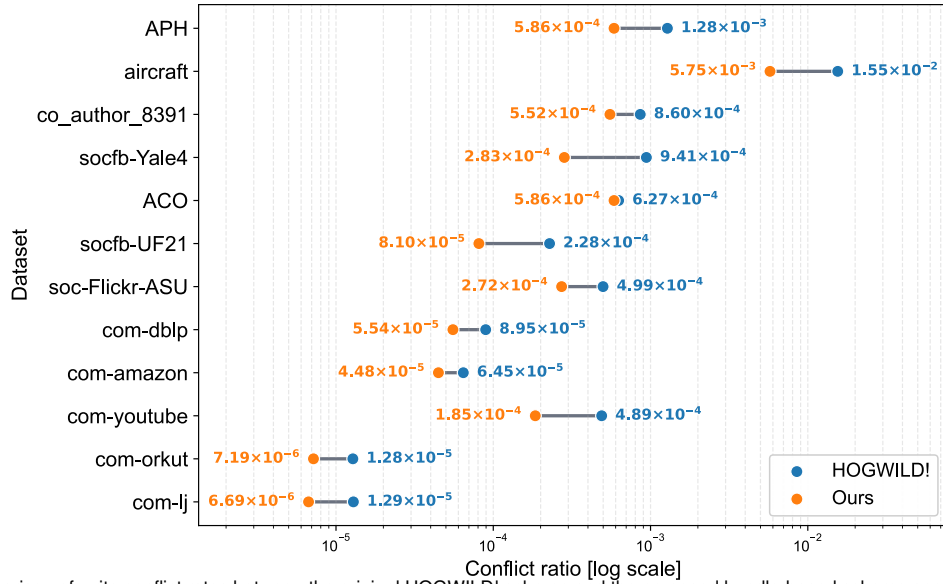


Fig. 3: Comparison of write conflict rates between the original HOGWILD! scheme and the proposed bundle-based scheme across all datasets.

2 EMPIRICAL COMPARISON OF CONFLICT RATES

This section empirically compares write conflict rates on all 12 benchmark datasets under two settings:

- the original HOGWILD! scheme [13], in which edges are randomly shuffled and evenly distributed across threads for independent processing;
- our bundle-based scheme, in which edges sharing the same source node are grouped into computational bundles and assigned to the same thread, as described in the submitted paper.

The implementation and all other optimization hyperparameters are identical besides the two settings. During layout optimization, an atomic counter is maintained for each node position in the position array $\mathbf{Y}$ to detect concurrent updates by multiple threads. The write conflict ratio is defined as

$$\text{Conflict Ratio} = \frac{N_{\text{conflict}}}{N_{\text{update}}},$$

where N_{conflict} denotes the total number of detected write conflicts and N_{update} denotes the total number of position updates. Each experiment is repeated five times, and the reported conflict rate is averaged over all runs to reduce the effect of randomness.

Figure 3 compares the write conflict rates under the two settings. The proposed bundle-based scheme consistently achieves lower conflict rates across all datasets. The largest reductions are observed on the *aircraft* and *com-youtube* datasets, where the conflict rate decreases by 63.04% and 62.20%, respectively. Using the coefficient of variation (CV) [16] of node degrees to measure the degree of imbalance, these two datasets have CV values of 6.6 and 9.6, respectively, far exceeding the typical range of 1.0–2.0 observed in most graphs. This result supports the claim made in the submitted paper that grouping edges by source node into computational bundles is particularly effective at reducing write conflicts in graphs with highly skewed degree distributions.

3 COMPLETE EXPERIMENTAL RESULTS

This section provides additional experimental results omitted from the submitted paper. We first present a detailed analysis of parameter selection and sensitivity, then report the complete visual quality heatmaps for all methods including their parallel variants, and finally provide the full runtime comparison.

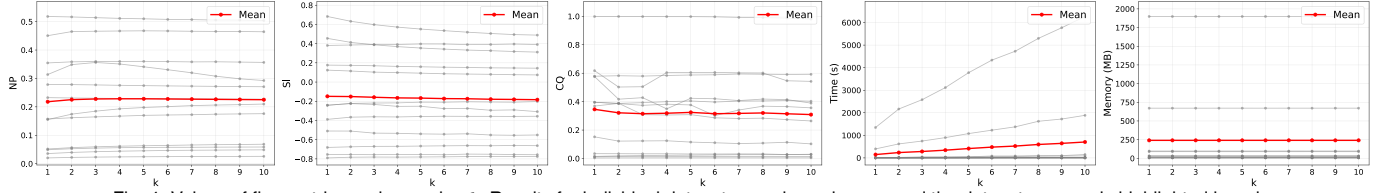


Fig. 4: Values of five metrics under varying k . Results for individual datasets are shown in gray, and the dataset average is highlighted in red.

Complete Metric Results under Varying k . To evaluate the robustness of the only algorithm-specific parameter introduced by SNAP-tFDP, we vary the number of negative samples k and report results for every dataset individually in Figure 4. Across graphs of different sizes and structures, the three visual quality metrics (NP, SI, and CQ) remain relatively stable as k changes, indicating that SNAP-tFDP does not require extensive parameter tuning. For efficiency metrics, runtime increases approximately linearly with k , while peak memory usage remains nearly constant, consistent with the theoretical time complexity of $|E|(1+k)$ and space complexity of $2(|V| + |E|)$. These results suggest that a single default value of k generalizes well across diverse graphs.

Complete Metric Results under Varying T . We evaluate the impact of the number of epochs T on all datasets and report representative results for small (*APH* with $|V| = 7,487$, $|E| = 119,043$), medium-sized (*socfb-UF21* with $|V| = 35,111$, $|E| = 1,465,654$), and large (*com-lj* with $|V| = 3,997,962$, $|E| = 34,681,189$) graphs in Figure 5. Across all graph scales, the three quality metrics (NP, SI, and CQ) improve rapidly during the first 50 epochs and then gradually stabilize, indicating that most layout refinement is completed early in the optimization process. The serial SNAP-tFDP algorithm and its two lock-free parallel variants exhibit nearly identical convergence behavior, with differences typically smaller than the run-to-run variability caused by stochastic negative sampling. These results demonstrate that the performance of SNAP-tFDP is relatively insensitive to the exact choice of T once a sufficient number of epochs is reached, and that a single default value ($T = 50$) generalizes well across graphs of different sizes without dataset-specific tuning.

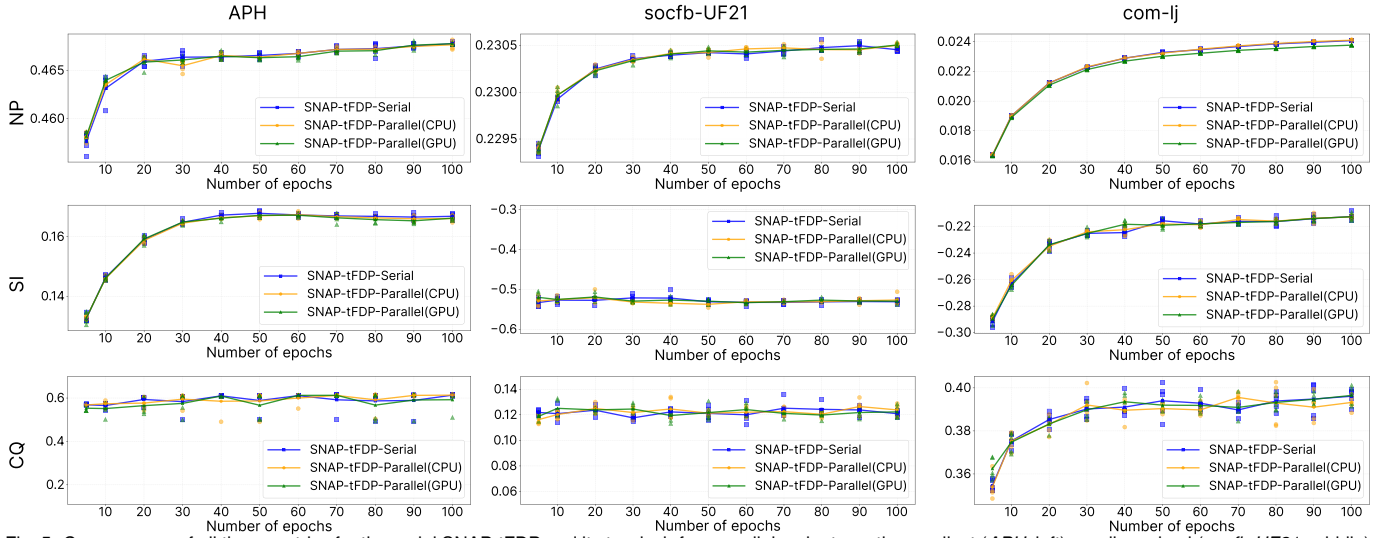


Fig. 5: Convergence of all three metrics for the serial SNAP-tFDP and its two lock-free parallel variants on the smallest (*APH*, left), medium-sized (*socfb-UF21*, middle), and largest (*com-lj*, right) graphs. Semi-transparent points correspond to five independent runs and demonstrate the stability of the parallel optimization process.

Complete Visual Quality Heatmaps. Figure 6 presents the complete visual quality results. Here, *-Parallel and *-GPU denote the CPU and GPU parallel variants, respectively. Overall, the parallel variants achieve visual quality consistent to their serial counterparts, indicating that parallelization has little impact on layout quality. SNAP-tFDP consistently attains the best average performance across all three metrics. In contrast, FA2-GPU exhibits noticeable quality variations relative to FA2, performing better on some datasets (e.g., *aircraft*) but worse on others (e.g., *socfb-Yale4*). Nevertheless, FA2-GPU remains substantially inferior to SNAP-tFDP across all three metrics.

Full Runtime Comparison. Figure 7 shows runtime as a function of the number of edges, additionally including four methods omitted from the submitted paper (FR, FA2, Linlog, and tsNET). The least-squares fitted lines indicate that all four methods scale substantially worse than the methods included in the submitted paper. Compared with SNAP-tFDP, the single-threaded CPU implementations of Linlog, FR, and FA2 are slower by 15743%, 16666%, and 54436% on average, respectively. While GPU-enabled tsNET exhibits a gentler fitted slope due to the limited number of available measurements, it is even 113477% slower than SNAP-tFDP-GPU on the datasets where it completes successfully. These results reinforce the efficiency and scalability of SNAP-tFDP.

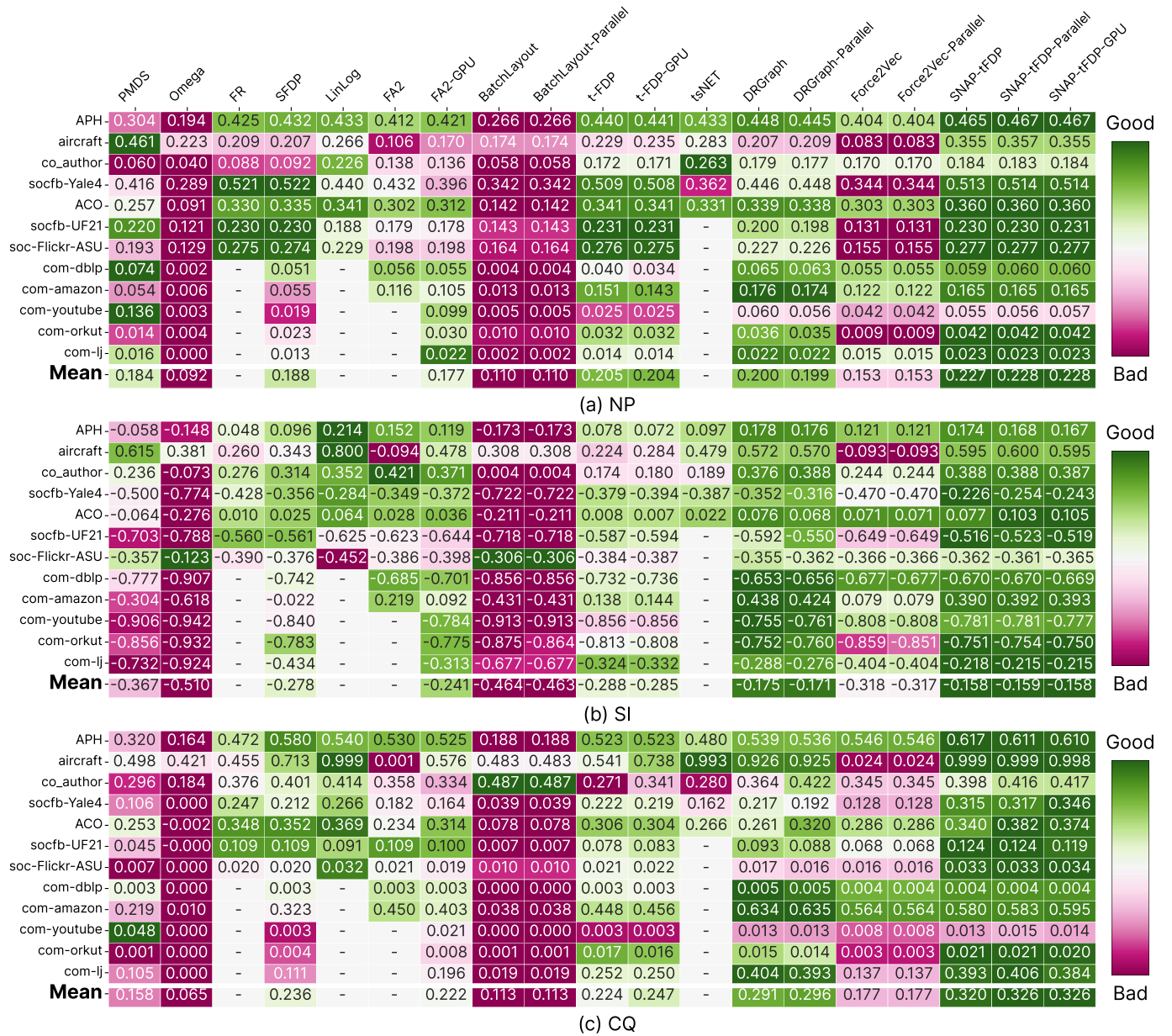


Fig. 6: Heatmaps employing a color-blind friendly pink-to-green colormap illustrate the scores of **NP** (a), **SI** (b), and **CQ** (c) for layouts generated by nine methods across all datasets. Empty cells indicate that the graph was too large to be processed by the corresponding method. Each row corresponds to a dataset, and each column to a layout method. Colors are scaled row-wise based on the best and worst within each dataset.

4 VISUAL RESULTS OF ALL GRAPH LAYOUT METHODS

In the end, we present the complete set of layouts produced by tested algorithms: **two stress-based methods** (PMDS [1] and Omega [10]), **six spring-electrical methods** (FR [4], SFDP [5], LinLog [9], FA2 [6] + FA2-GPU [2], BatchLayout [11], and t-FDP [17]), and **three dimensionality reduction-based methods** (tsNET [7], DRGraph [18], and Force2Vec [12]), along with our SNAP-tFDP, its two parallel variants, across all datasets (Table 2 in the submitted paper).

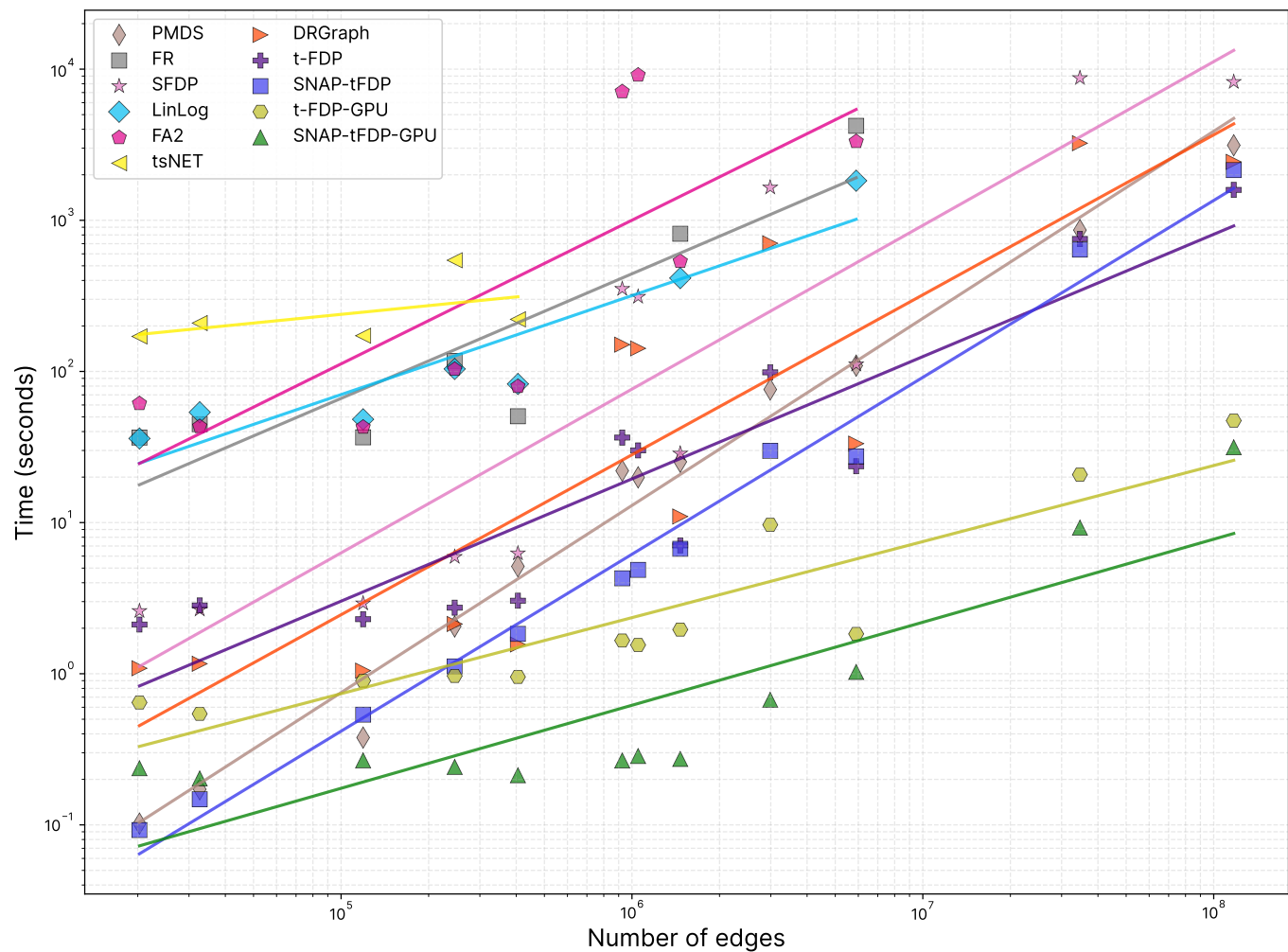
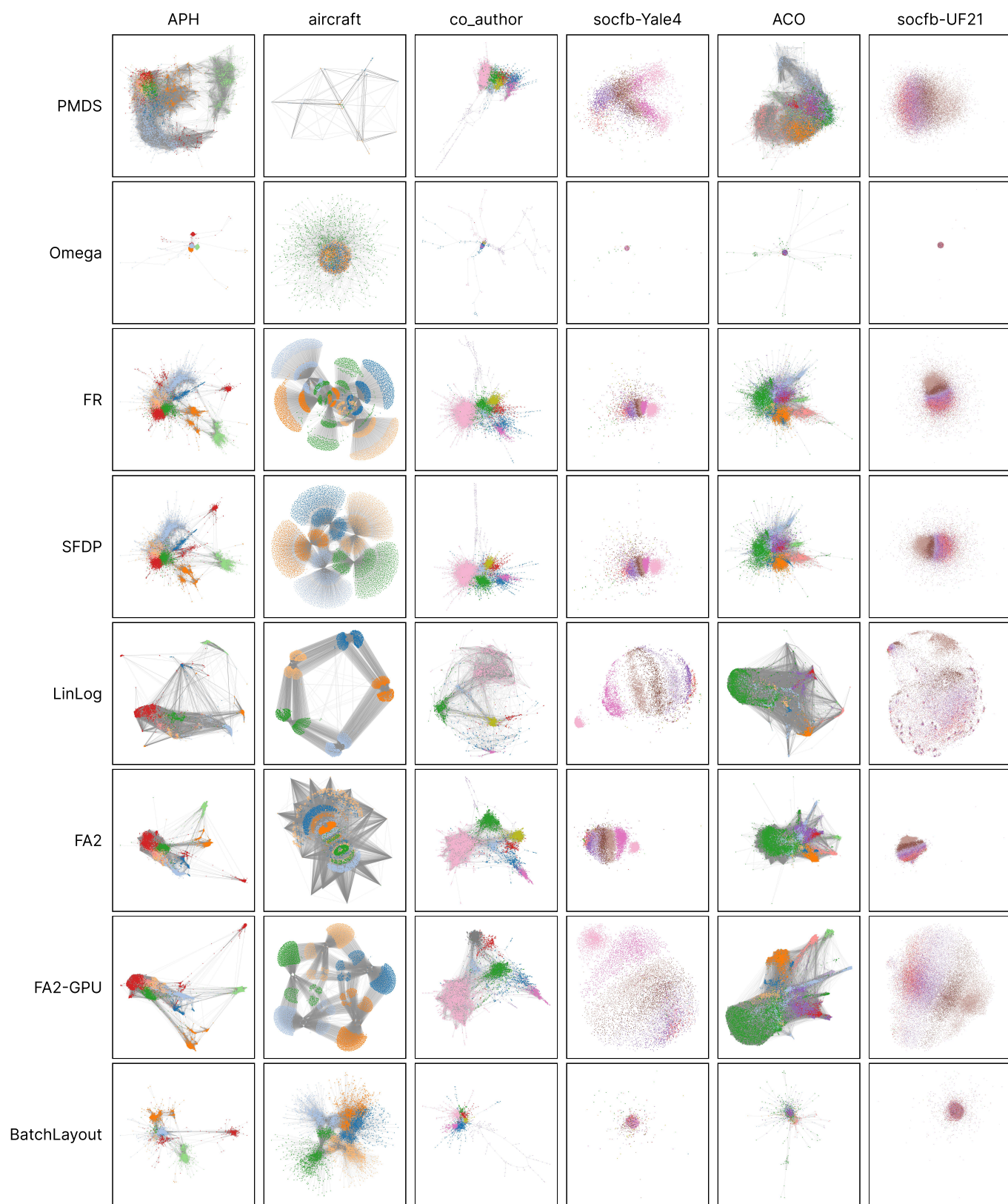
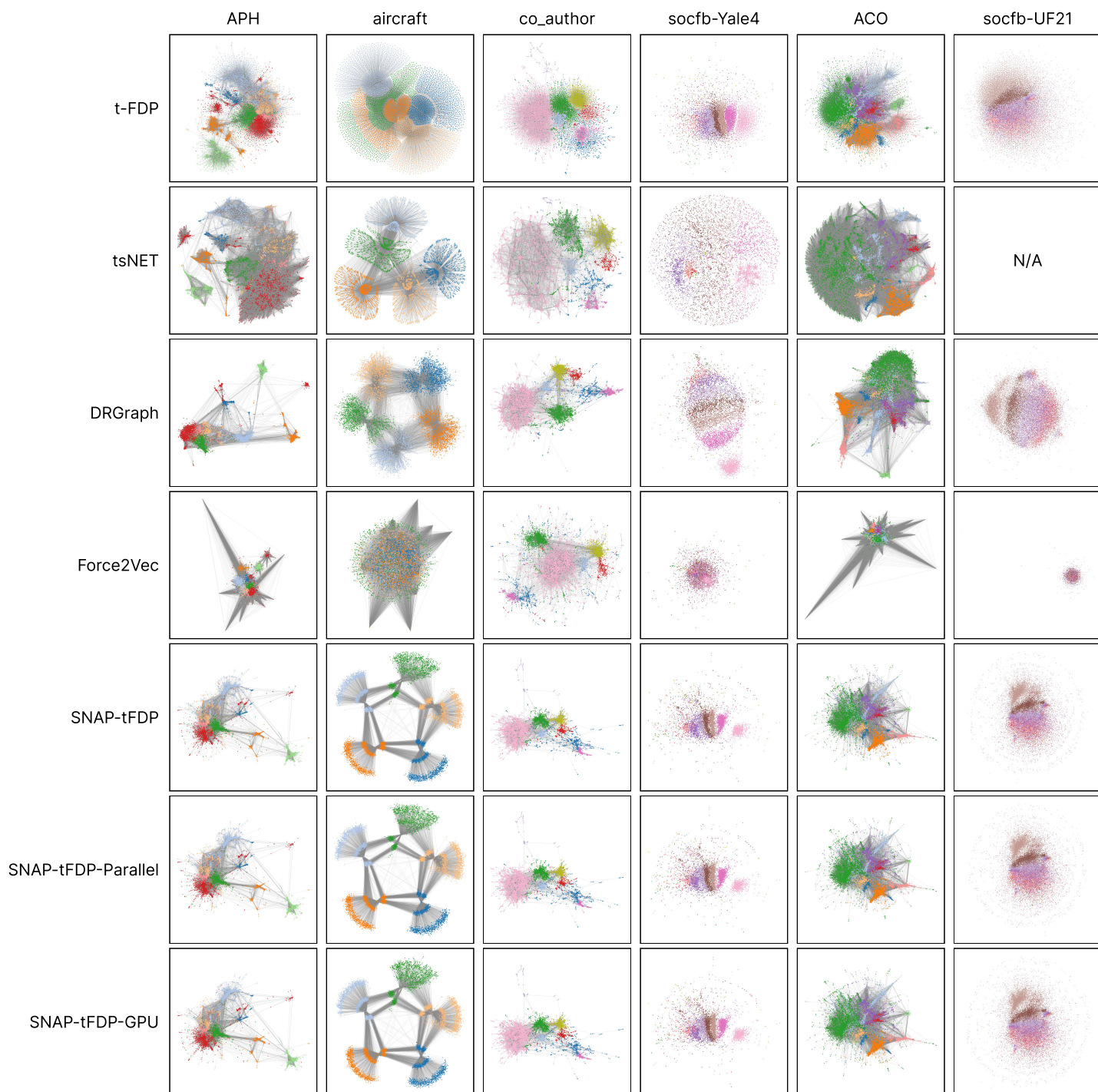
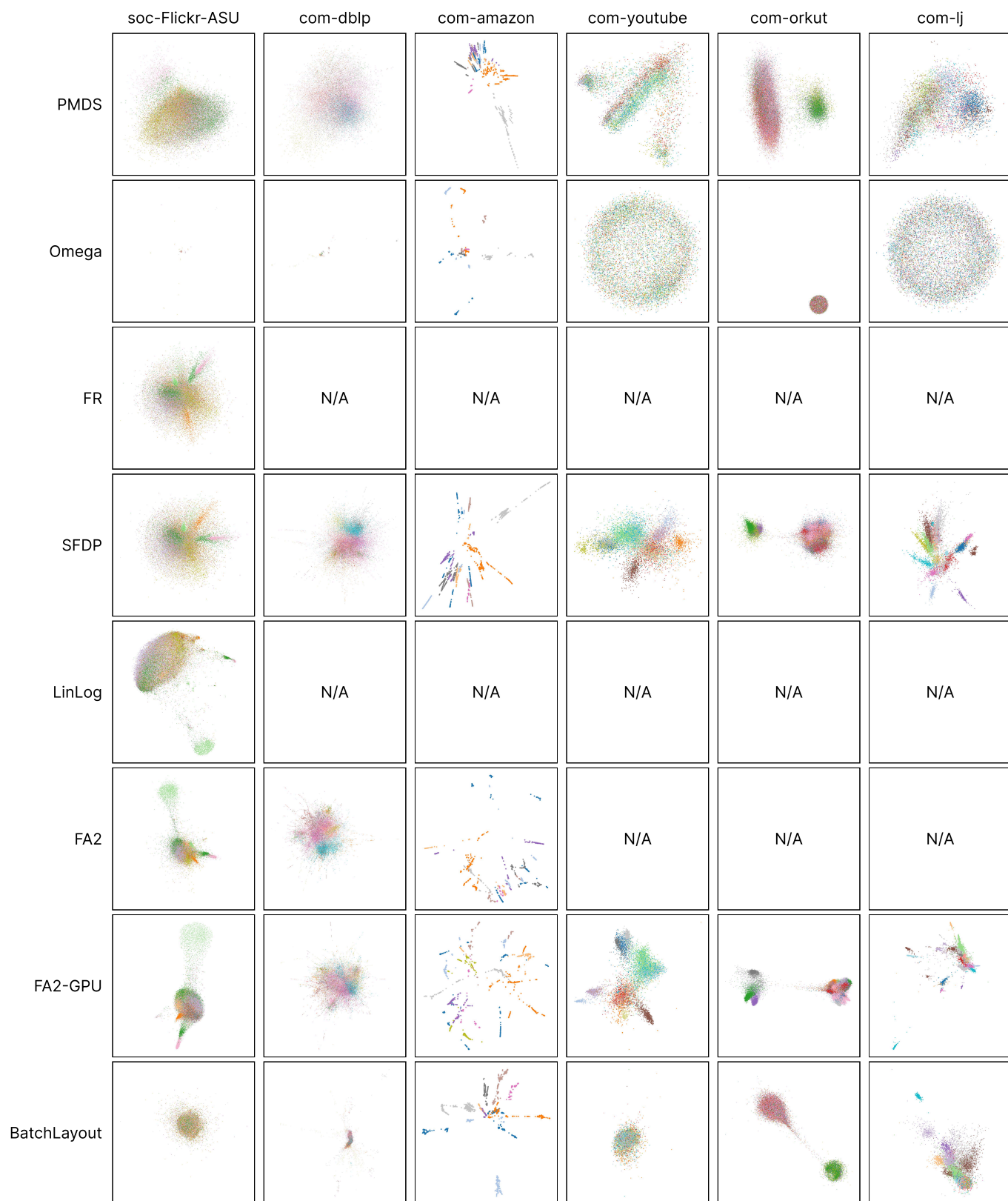


Fig. 7: Runtime of all twelve layout methods, together with the GPU implementations of FA2, t-FDP, and SNAP-tFDP, across all datasets.









REFERENCES

- [1] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *International Symposium on Graph Drawing*, pp. 42–53. Springer, 2006. doi: [10.1007/978-3-540-70904-6_6](https://doi.org/10.1007/978-3-540-70904-6_6) 5
- [2] A. Fender, B. Rees, and J. Eaton. Rapids cugraph. In *Massive Graph Analytics*, pp. 483–493. Chapman and Hall/CRC, 2022. 5
- [3] E. Frøkjær, M. Hertzum, and K. Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*, pp. 345–352. ACM, 2000. doi: [10.1145/332040.332455](https://doi.org/10.1145/332040.332455) 1
- [4] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. doi: [10.1002/spe.4380211102](https://doi.org/10.1002/spe.4380211102) 5
- [5] Y. F. Hu. Efficient and high quality force-directed graph drawing. *The Mathematica Journal*, 10:37–71, 2005. 5
- [6] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014. doi: [10.1371/journal.pone.0098679](https://doi.org/10.1371/journal.pone.0098679) 5
- [7] J. F. Krüger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. Graph layouts by t-sne. *Computer Graphics Forum*, 36(3):283–294, 2017. doi: [10.1111/cgf.13187](https://doi.org/10.1111/cgf.13187) 5
- [8] A. Meidiana, S.-H. Hong, P. Eades, and D. Keim. A quality metric for visualization of clusters in graphs. In *Graph Drawing and Network Visualization: 27th International Symposium, GD 2019, Prague, Czech Republic, September 17–20, 2019, Proceedings*, pp. 125–138. Springer-Verlag, Berlin, Heidelberg, 2019. doi: [10.1007/978-3-030-35802-0_10](https://doi.org/10.1007/978-3-030-35802-0_10) 2
- [9] A. Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007. doi: [10.7155/jgaa.00154](https://doi.org/10.7155/jgaa.00154) 5
- [10] Y. Onoue. Graph drawing stress model with resistance distances. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–10, 2026. doi: [10.1109/TVCG.2026.3694437](https://doi.org/10.1109/TVCG.2026.3694437) 5
- [11] M. K. Rahman, M. H. Sujon, and A. Azad. Batchlayout: A batch-parallel force-directed graph layout algorithm in shared memory. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 16–25. IEEE, 2020. doi: [10.1109/PacificVis48177.2020.3756](https://doi.org/10.1109/PacificVis48177.2020.3756) 5
- [12] M. K. Rahman, M. H. Sujon, and A. Azad. Force2vec: Parallel force-directed graph embedding. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 442–451. IEEE, 2020. doi: [10.1109/ICDM50108.2020.00053](https://doi.org/10.1109/ICDM50108.2020.00053) 5
- [13] B. Recht, C. Re, S. Wright, and F. Niu. HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds., *Advances in Neural Information Processing Systems*, vol. 24. Curran Associates, Inc., 2011. 3
- [14] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: [10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7) 2
- [15] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 2
- [16] Wikipedia. Coefficient of variation. https://en.wikipedia.org/wiki/Coefficient_of_variation, 2026. Accessed: 2026-06-29. 3
- [17] F. Zhong, M. Xue, J. Zhang, F. Zhang, R. Ban, O. Deussen et al. Force-directed graph layouts revisited: a new force based on the t-distribution. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3650–3663, 2023. doi: [10.1109/TVCG.2023.3238821](https://doi.org/10.1109/TVCG.2023.3238821) 1, 2, 5
- [18] M. Zhu, W. Chen, Y. Hu, Y. Hou, L. Liu, and K. Zhang. DRGraph: An efficient graph layout algorithm for large-scale graphs by dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1666–1676, 2020. doi: [10.1109/TVCG.2020.3030447](https://doi.org/10.1109/TVCG.2020.3030447) 5