AutoFDP: A Criteria-driven Force-based Framework for Graph Layouts

Mingliang Xue, Yifan Wang, Zhi Wang, Lifeng Zhu, Lizhen Cui,
Yueguo Chen, Zhiyu Ding, Oliver Deussen, Yunhai Wang

Abstract—We provide more details of our evaluation results in this supplemental material. Section 1 reports the statistics of the datasets we used and the common criteria that we use to drive the automated selection of layouts and measure the layout quality. Section 2 shows the runtime and visual results of our experiments that optimize every single criterion of the nine criteria described in the SGD^2 paper and six sums of weighted criteria by our method and SGD^2 obtained from the experiments in Section V-A of the main paper. Section 3 shows the visual results obtained by optimizing four criteria using our method in comparison to the layouts generated by four classic methods, which also corresponds to the experiments in Section V-A of the main paper. Lastly, Section 4 shows the comparison between our method and deep learning-based methods in optimizing six different criteria, which corresponds to the experiments in Section V-B of the main paper.

Index Terms—Graph Layout, Readability Criteria, Optimization

1 Datasets and Graph Readability Criteria

1.1 Datasets

First, we provide details of all experimental datasets. For the comparison to SGD^2 [1] and classic methods, we utilized 45 synthetic graphs, including 15 grid graphs, 15 binary trees, and 15 clustered graphs, along with 30 real graphs. For the comparison to DeepGD [2] and SmartGD [3], we employed 150 small synthetic graphs, comprising 50 grid graphs, 50 binary trees, and 50 clustered graphs, as well as 1000 graph datasets selected from the Rome dataset.

Scatter plots in Figure 1 showcase the distribution of the number of nodes and edges for the synthetic graphs. Figure 1a shows the distribution of the number of nodes and edges for 45 synthetic graphs and Figure 1b for 150 small synthetic graphs used in Sections 5.1 and 5.2 of the main paper, respectively. Figure 2 illustrates the distribution of the number of nodes and edges for the 1000 small graphs selected from the Rome dataset. Additionally, Table 1 provides a detailed description of the 30 real graphs.

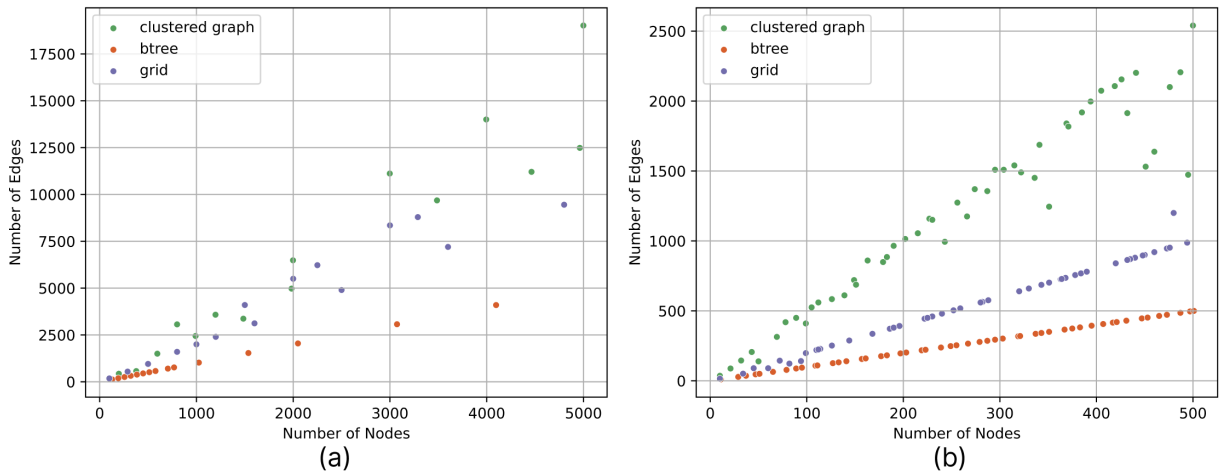

Fig. 1: Scatter plots showing the distribution of the number of nodes and edges for (a) 45 synthetic graphs and (b) 150 small synthetic graphs used in Sections 5.1 and 5.2 of the main paper, respectively.

1.2 Graph Readability Criteria

Next, we provide a brief overview of the aesthetic criteria used to drive the automated selection of layouts and assess the layout quality in the experiments. We use the same quality measures for all the methods in our experiments while different methods employ different loss functions. The loss functions employed by SGD^2 and DeepGD were described in their papers [1, 2]. For AutoFDP, the same formulas used to measure layout quality were also utilized as loss functions to optimize most of the aesthetic objectives. However, to enhance optimization efficiency for NR, AR, and GP objectives, we incorporated the loss function employed by SGD^2.

Normalized Stress Error (SE) [11]. To assess the distance-preservation of a layout, the normalized stress error measures the squared difference for the shortest path distance d_{ij} between nodes i and j and their Euclidean distance:

$$\mathrm{Q}_{SE} = \mathrm{L}_{SE} = \frac{2}{|V|(|V|-1)} \sum_{i<j} \frac{(\bar{s}\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})^2}{{d_{ij}}^2}. \tag{1}$$

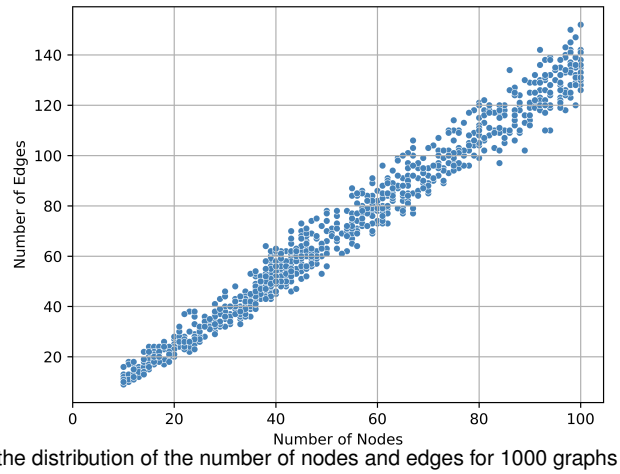

Fig. 2: Scatter plot showing the distribution of the number of nodes and edges for 1000 graphs selected from the Rome dataset.

Table 1: The statistics of 30 real graphs.

	Nodes	Edges	Description and References
lesmis	77	254	co-appearances of characters in Les Miserables [4,5]
rajat11	135	377	Circuit simulation problem [6]
jazz	198	2742	Jazz muscicians network [6]
visbrazil	222	336	collaboration [4,5]
mesh3e1	289	800	structural problem [4,5]
netscience	379	914	co-author network [4,5]
dwt_419	419	1572	Structural problem [6]
congress	475	10222	Twitter interaction network for the 117th United States Congress [6]
bfwa782	782	3394	structural [7]
qh882	882	1533	Power network problem [6]
soc-wiki-Vote	889	2914	Wikipedia who-votes-on-whom network [6]
cage8	1015	4994	DNA electrophoresis result [4,5]
circuit204	1020	3973	Circuit simulation problem [6]
ca-CSphd	1025	1043	PhD's in computer science [6]
bcsstk09	1083	8677	mesh-like, structural [4,5]
bus1138	1138	1458	power network problem [8]
fpga_dcop1220	1220	2810	circuit simulation matrix [8]
qh1484	1470	2475	power network problem [9]
bcspwr07	1612	2106	power network problem [8,9]
utm1700b	1700	14626	Electromagnetics problem [6]
bio-CE-HT	2194	2688	Biological Network [6]
data	2851	15093	Miscellaneous Network [6]
ca-GrQc	4158	13422	Collaboration network of Arxiv General Relativity [6]
EVA	4475	4652	Pajek network [6]
3elt	4720	13722	2D/3D Problem [4,5,10]
USpowerGrid	4941	6594	power network problem [4,5,10,11]
add32	4960	9462	electronic circuit matrix [10,12,13]
p2p-Gnutella08	6299	20776	Gnutella peer to peer network from August 2002 [6]
lastfm_asia	7624	27806	Social network of LastFM users from Asian countries [6]
CA-HepTh	8638	24806	Collaboration network of Arxiv High Energy Physics Theory [6]

where $\bar{s}$ is a weight factor to scale the layout [14,15]. When the nodes i and j are in different components (a graph has multiple components), we only compute stress errors for node pairs with finite distances.

Ideal Edge Length (IL). Assuming the ideal edge length for all edges would be ℓ, IL computes the variance from this ideal length:

$$\mathrm{Q}_{IL} = \mathrm{L}_{IL} = \frac{1}{|E|} \sum_{(i,j)\in E} \frac{(||\mathbf{x}_i - \mathbf{x}_j|| - \ell)^2}{\ell^2}, \tag{2}$$

where ℓ is 1 by default.

Neighborhood Preservation (NP). The neighborhood preservation measures the alignment between the input graph and the k-nearest neighborhood graph defined in the layout with the Jaccard index. For each node i, we define

$$k_i = |NG(i,r)|, \quad \mathrm{Q}_{NP_i} = \mathrm{L}_{NP_i} = 1 - \frac{NG(i,r) \cap NL(\mathbf{x}_i, k_i)}{NG(i,r) \cup NL(\mathbf{x}_i, k_i)}, \tag{3}$$

where $NG(i,r)$ is the set of r-ring neighboring nodes in data space, and and $NL(\mathbf{x}_i, k_i)$ are the k_i-nearest-neighbors of $\mathbf{x}_i$ in layout space. Here, we use $r = 2$ and normalize the sum of all NP_i to [0,1].

Crosslessness (CL). We use the crosslessness metric [16] to quantify the amount of edge crossings:

$$\mathrm{Q}_{CL} = \mathrm{L}_{CL} = c/c_{max},$$

where c is the number of edge crossings, and c_{max} is the maximal number in each graph. A smaller value indicates fewer edge crossings.

Crossing Angle (CA). Since the maximum edge crossing angle is $\frac{\pi}{2}$, CA measures the absolute discrepancy between each crossing angle and the target crossing angle of $\frac{\pi}{2}$:

$$\mathrm{Q}_{CA} = \mathrm{L}_{CA} = \frac{1}{|CE|}\sum_{i,j\in CE}\left|\theta_{i,j} - \frac{\pi}{2}\right| / \frac{\pi}{2}, \tag{4}$$

where CE is the set of crossing edges, $\theta_{i,j}$ is the crossing angle of edges i and j. A small CA indicates large crossing angles between edges, resulting in better readability of the graph layout.

Minimum Angle (MA). For the node i, MA is defined as the deviation between the minimum angle between incident edges $\theta_{min}(i)$ and the ideal minimum angle $\theta(i)$:

$$\theta(i) = \frac{2\pi}{\mathrm{degree}(i)},\ \mathrm{Q}_{MA} = \mathrm{L}_{MA} = \frac{1}{|V|}\sum_{i}^{|V|}\frac{|\theta(i) - \theta_{min}(i)|}{\theta(i)}, \tag{5}$$

where MA is in the range [0,1].

Node Resolution (NR). To assess the amount of node overlap, NR is usually defined as the minimum distance between two nodes in a layout [17].

$$d_{max} = \max_{i,j\in V}||\mathbf{x}_i - \mathbf{x}_j||, \mathrm{Q}_{NR} = 1 - \min(1, \frac{\min_{i,j\in V}||\mathbf{x}_i - \mathbf{x}_j||}{\sqrt{|V|}d_{max}}), \tag{6}$$

where d_{max} represents the distance between the two nodes with the farthest Euclidean distance in the layout. The loss function is defined as the sum of the differences between the distances of each node pair and d_{max}:

$$\mathrm{L}_{NR} = \sum_{i,j\in V, i\neq j}\max(0, (1 - \frac{||\mathbf{x}_i - \mathbf{x}_j||}{\sqrt{|V|}d_{max}})^2). \tag{7}$$

Aspect Ratio (AR). To indicate to what extent the bounding box of a layout approaches a square, we define it as:

$$\mathrm{Q}_{AR} = 1 - \min_{\theta\in\frac{2\pi k}{N}, k\in[0,\ldots,(N-1)]}\frac{\min(w_\theta, h_\theta)}{\max(w_\theta, h_\theta)}, \tag{8}$$

where w and h are the width and height of the bounding box. The quality measure for AR involves rotating the layout N times and measuring the minimum ratio between the width w_θ and height h_θ of the bounding box obtained in each rotation, we take $N = 7$ in our experiments. The loss function is defined as the cross-entropy of the ratio between the width w and height h of the bounding box:

$$\mathrm{L}_{AR} = -r log(\frac{w}{h}) - (1-r) log(1 - \frac{w}{h}). \tag{9}$$

Gabriel Graph Property (GP). For any edge l from the layout, we can draw a circle with the edge as its diameter. If all these circles contain no other nodes of the layout, then the graph is a Gabriel graph [18]. GP we call the proportion of edges without nodes in their circles [1].

$$\mathrm{Q}_{GP} = 1 - \min(1, \min_{(i,j)\in E, k\in V}\frac{||\mathbf{x}_k - \mathbf{c}_{ij}||}{r_{ij}}), \tag{10}$$

where $r_{i,j}$ and $\mathbf{c}_{i,j}$ are the radius and the midpoint of edge (i,j). The loss function is defined as the sum of the exceeding distances beyond $r_{i,j}$ for each edge to all other nodes.

$$\mathrm{L}_{GP} = \sum_{(i,j)\in E, k\in V, k\neq i,j}\max(0, r_{i,j} - ||\mathbf{x}_k - \mathbf{c}_{i,j}||)^2. \tag{11}$$

T-distributed Stochastic Neighbor Embedding (TSNE). The criterion TSNE aims at minimizing the divergence between the graph space similarity p_{ij} and the graph space similarity p_{ij}:

$$\mathrm{Q}_{TSNE} = \mathrm{L}_{TSNE} = \sum_{i\neq j} p_{ij} log\frac{p_{ij}}{q_{ij}}, \tag{12}$$

The formulas of graph space similarity p_{ij} and the graph space similarity q_{ij} of node i and j are as follows:

$$p_{ij} = p_{ji} = \frac{p_{i|j} + p_{j|i}}{2N}, p_{j|i} = \frac{exp(-\frac{d_{ij}^2}{2\sigma_i^2})}{\sum_{k\neq i} exp(\frac{d_{ik}^2}{2\sigma_i^2})}, \tag{13}$$

$$q_{ij} = q_{ji} = \frac{(1 + ||\mathbf{x}_i - \mathbf{x}_j||^2)^{-}1}{\sum_{k\neq l}(1 + ||\mathbf{x}_k - \mathbf{x}_l||^2)^{-}1}, \tag{14}$$

where d_{ij} indicates the graph distance between node i and j, and σ_i is a hyper parameter representing the Gaussian standard deviation for node i.

2 Comparison to SGD^2

Then we compare our method with SGD^2 by optimizing every single criterion of the nine criteria and six sums of weighted criteria provided by SGD^2 paper. The optimized single criterion including Stress Error (SE), Ideal Edge Length (IL), Neighborhood Preservation (NP), Crosslessness (CL), Minimum Angle (MA), Crossing Angle (CA), Node Resolution (NR), Aspect Ratio (AR) and Gabriel Graph Property (GP). The optimized sums of weighted criteria include SE+IL, SE+NP, SE+CL, SE+CA, SE+GP, and NP+CL.

We show the runtime comparison between our method and SGD^2 of optimizing nine single criteria and six multiple criteria in Figure 3. Also, we show the visual results of our experiments, where Figure 4 shows the visual results of optimizing each single criterion, and Figure 5 shows the visual results of optimizing multiple criteria on synthetic graphs. Due to their overwhelming number of visual results, we only selected

9 data with different structures for presentation. Among them are two clustered graphs, three binary tree graphs, and four grid graphs. In Figure 6 and Figure 7, we show the visual results of optimizing each single criterion and multiple criteria on 30 real graphs, respectively.

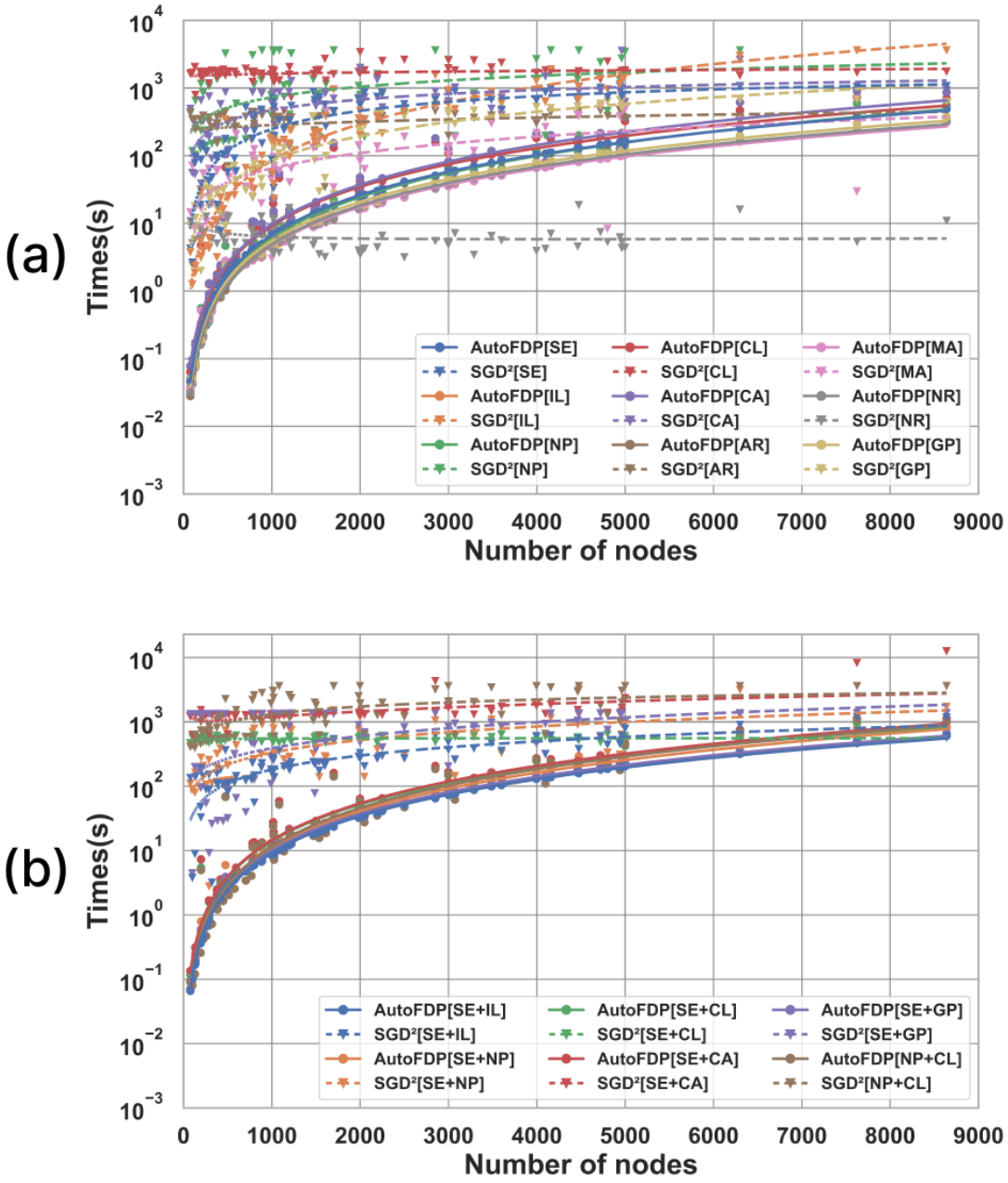

Fig. 3: Runtime comparison between our method and SGD^2: optimizing nine single criteria (a) and six multiple criteria (b).

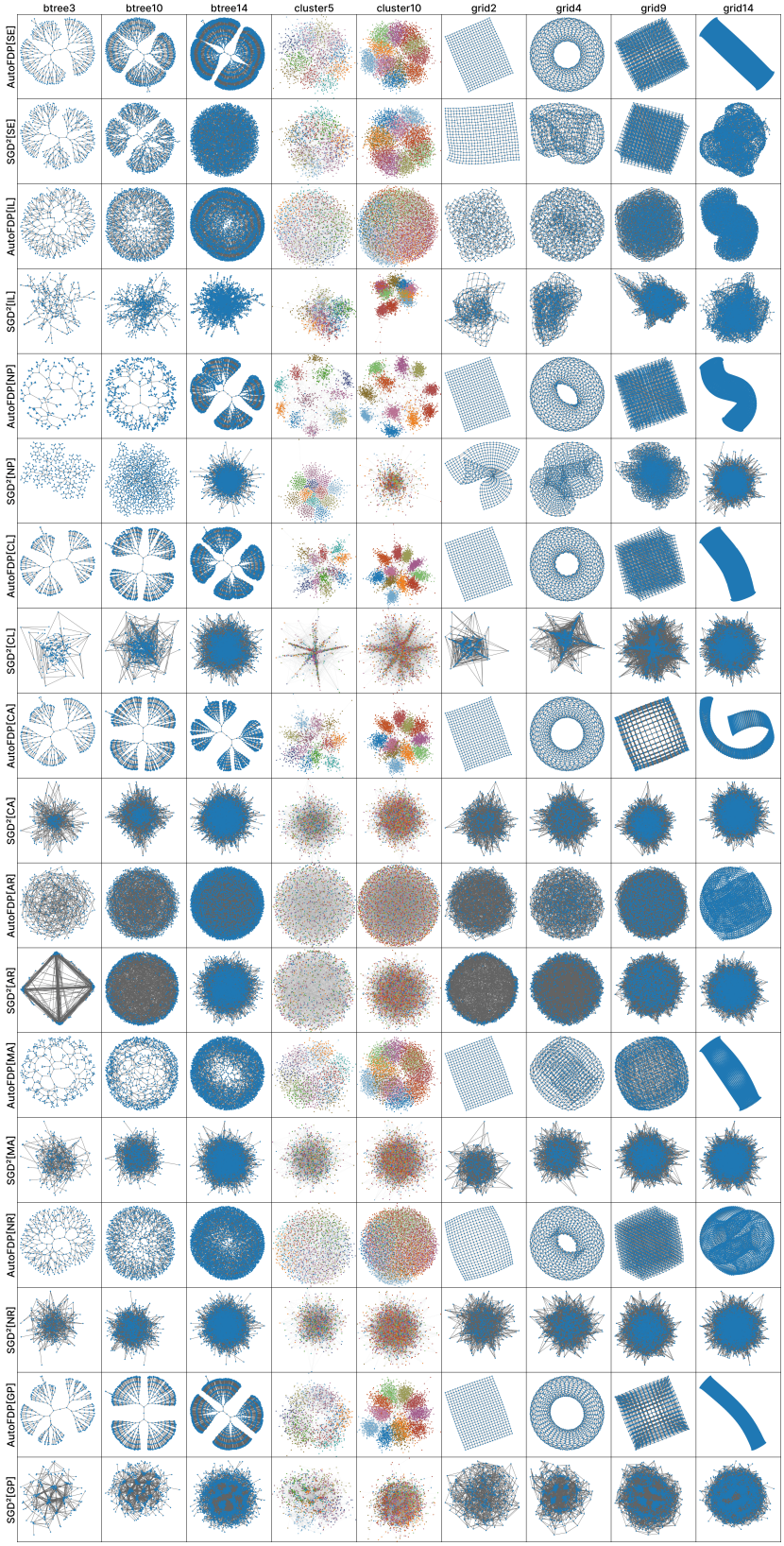

Fig. 4: Visual results generated by optimizing every single criterion on synthetic graphs.

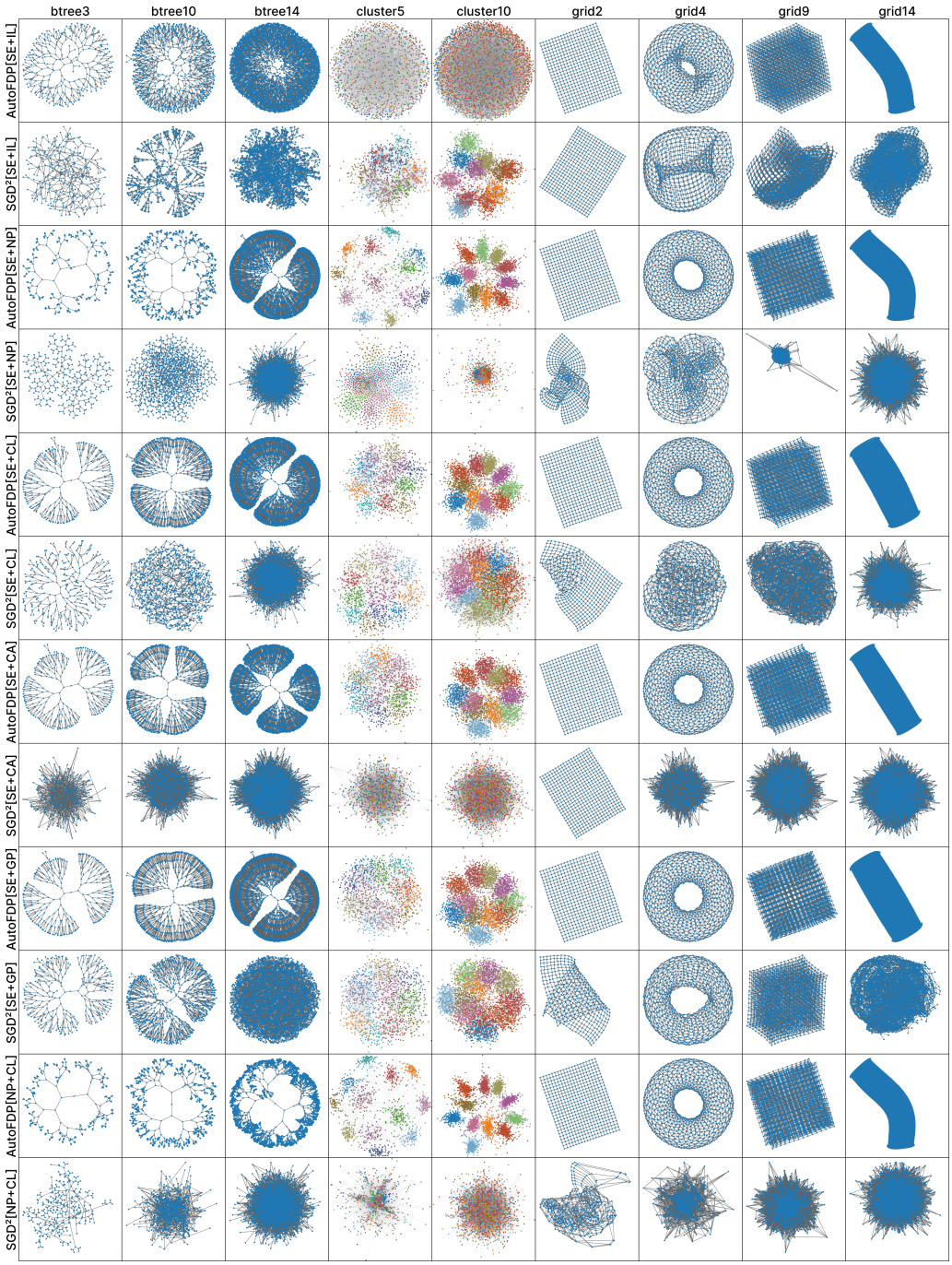

Fig. 5: Visual results generated by optimizing multiple criteria on synthetic graphs.

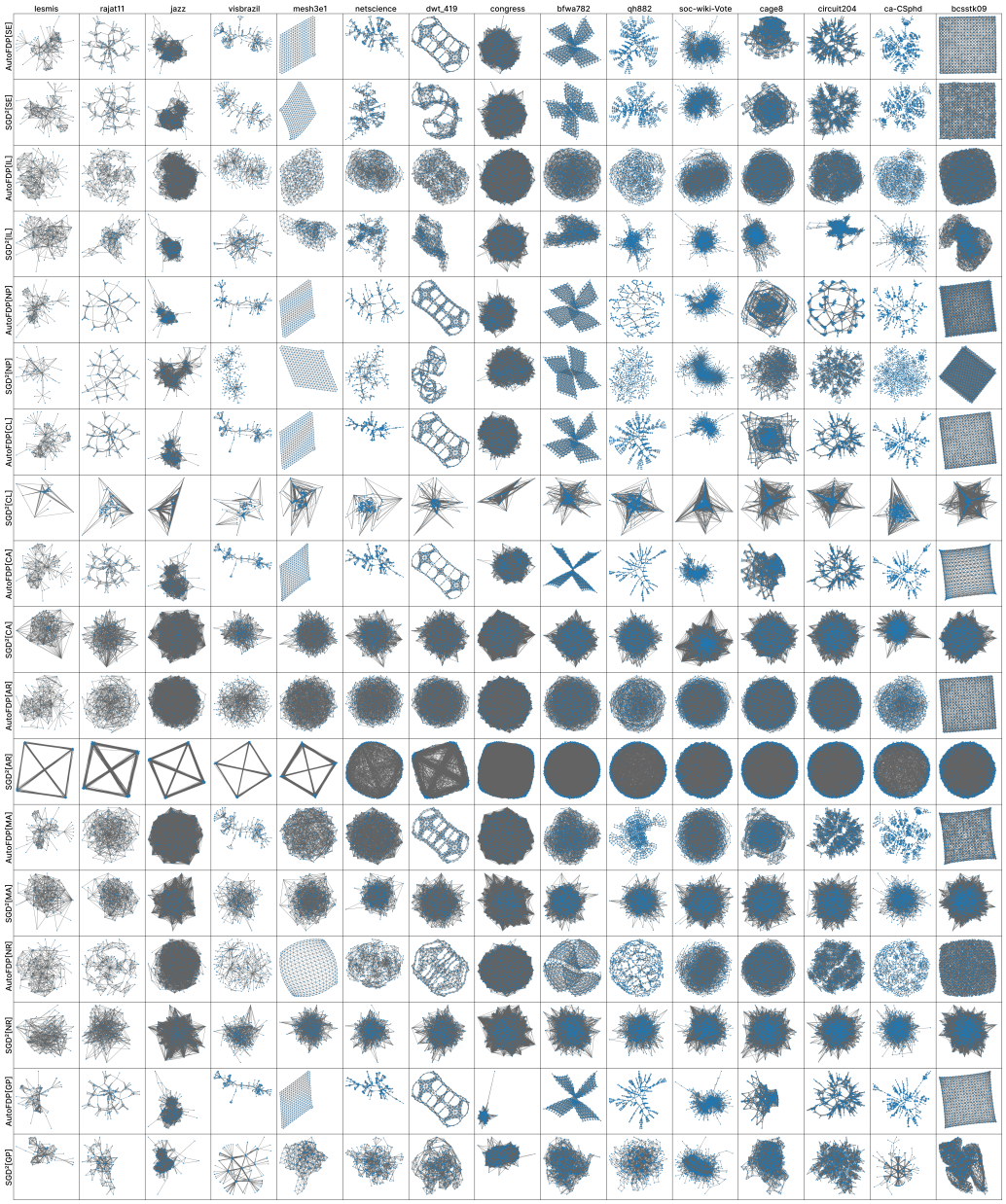

Fig. 6: Visual results generated by optimizing every single criterion on the first 15 real graphs.

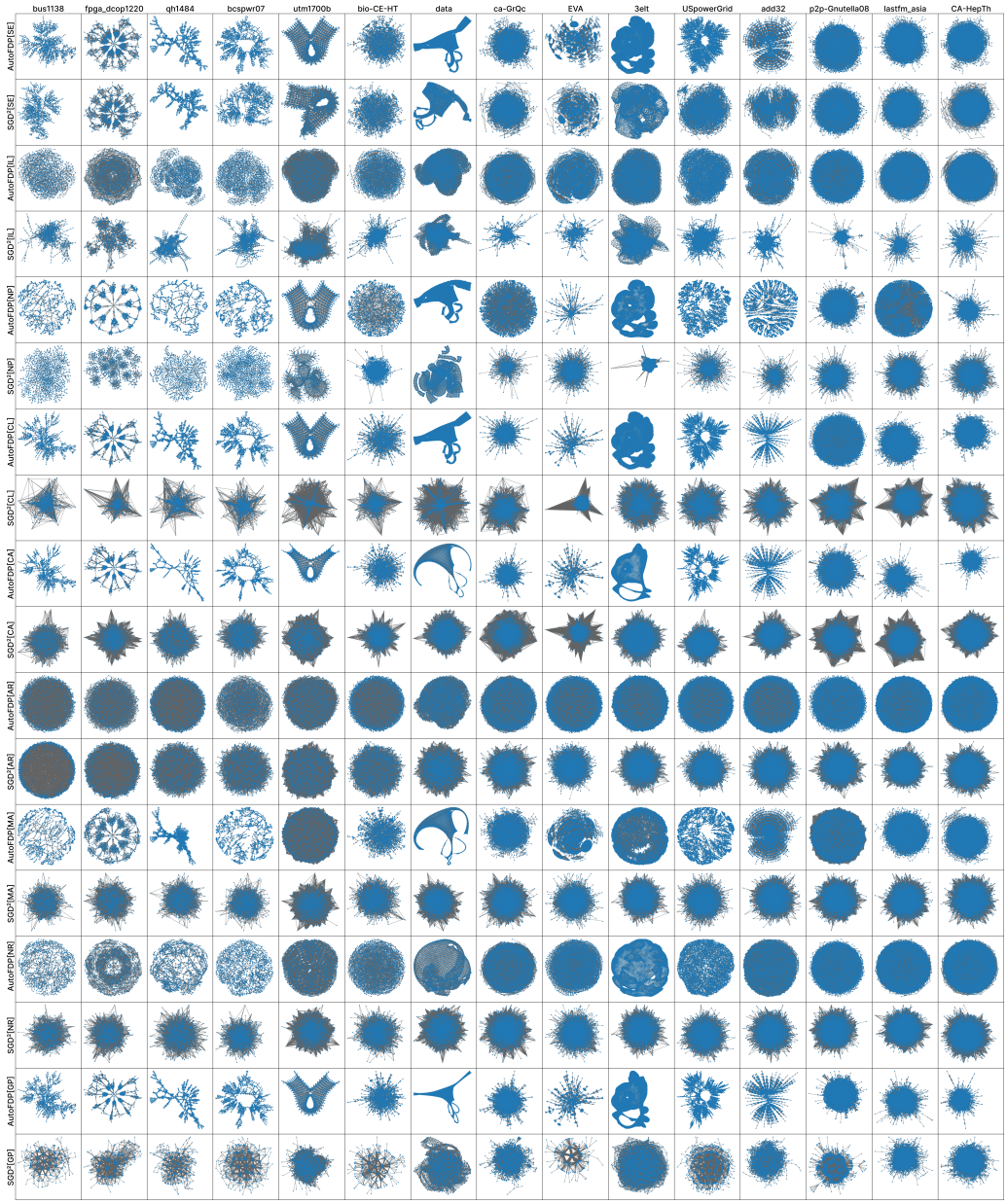

Fig. 7: Visual results generated by optimizing multiple criteria on the last 15 real graphs.

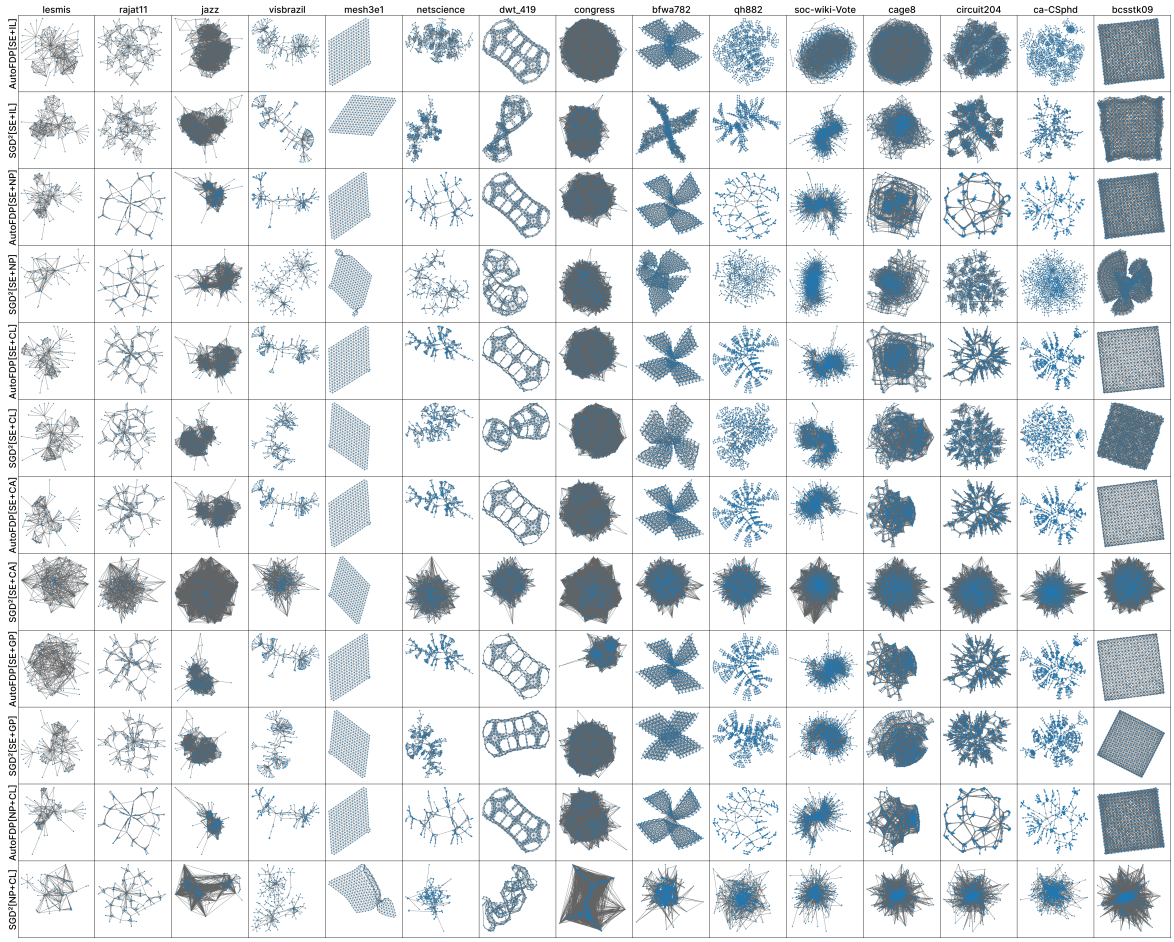

Fig. 8: Visual results generated by optimizing every single criterion on the first 15 real graphs.

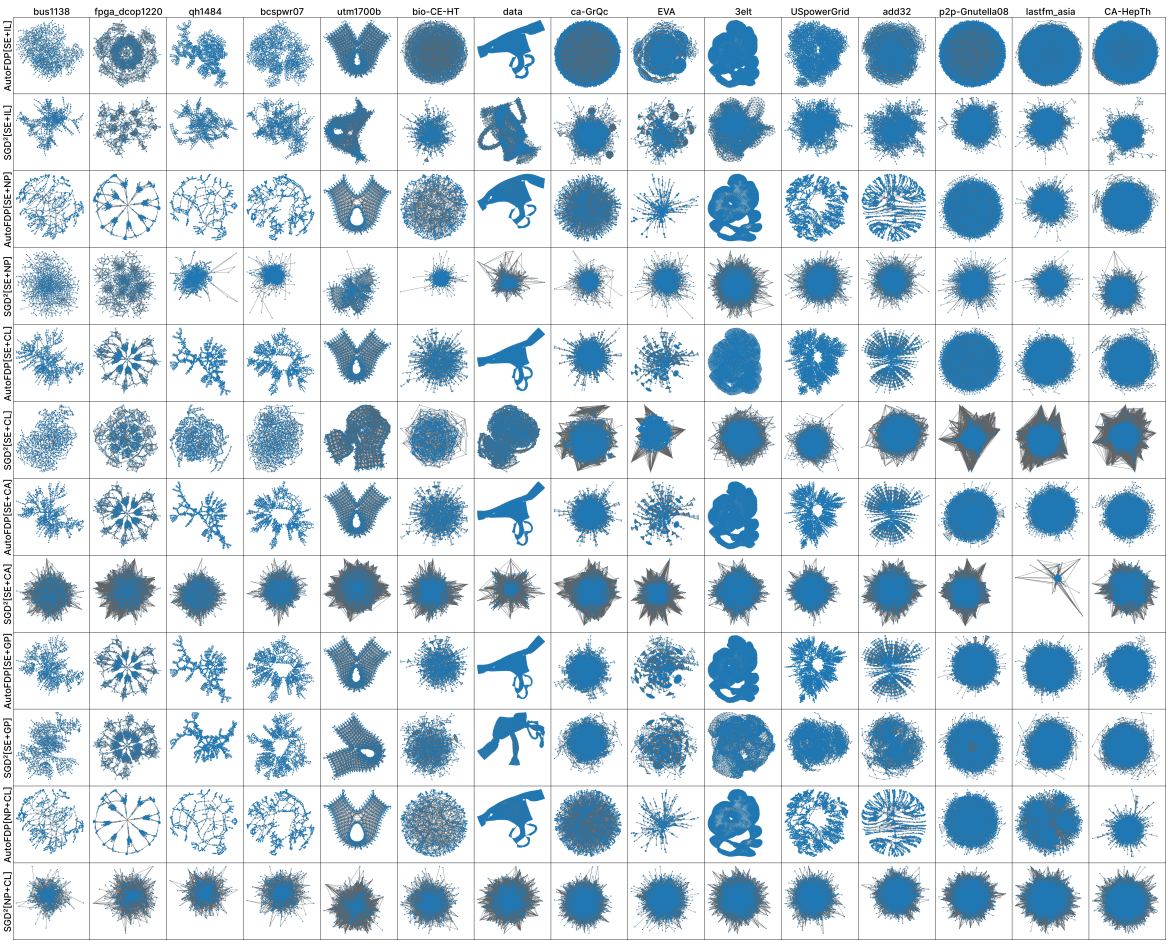

Fig. 9: Visual results generated by optimizing multiple criteria on the last 15 real graphs.

3 Comparison to Classic Methods

We further compared five kinds of classic methods (stress model [19], SFDP [12], Maxent [11], tsNET [5] and DRGraph [4]) with our method. We focus on the layout results obtained by optimizing a select set of criteria including Stress Error (SE), Neighborhood Preservation (NP), Crosslessness (CL), and their combinations by our method in this comparison. Figure 10, Figure 11, Figure 12, Figure 13, and Figure 14 show the visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL], and five classic methods on grids, binary trees, clustered graphs, and real graphs.

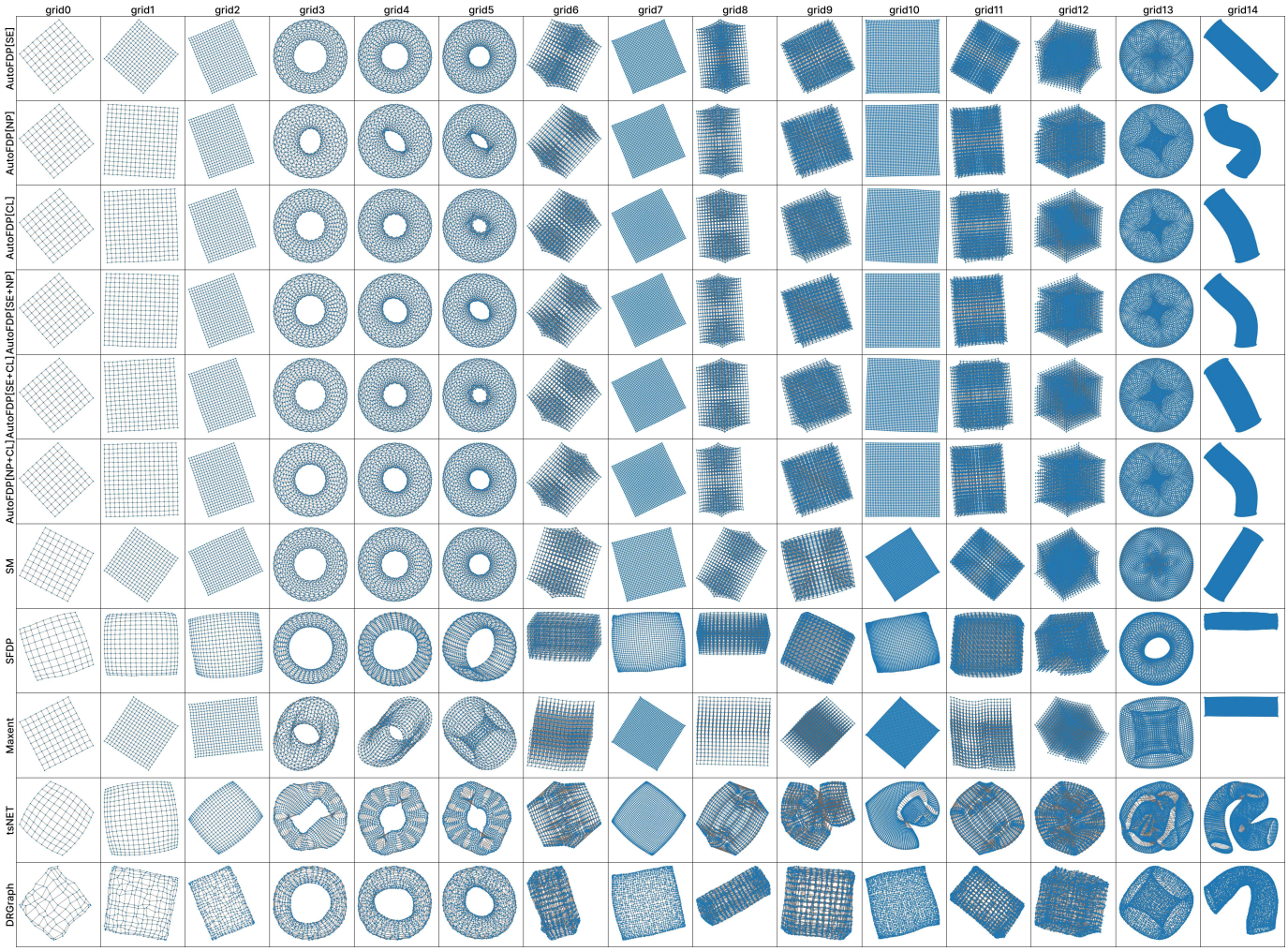

Fig. 10: Visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL] (top), and five classic methods (bottom) on grids.

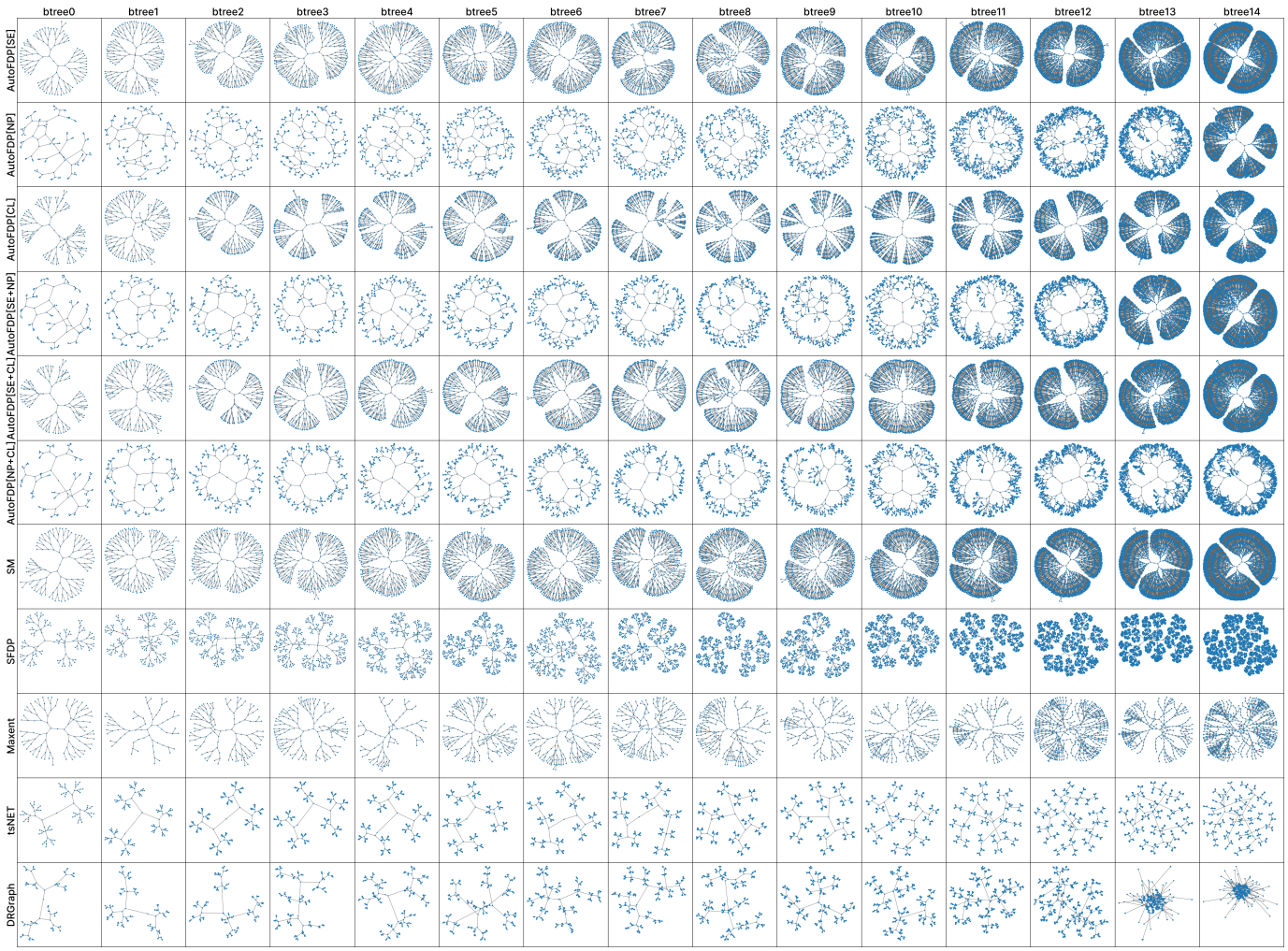

Fig. 11: Visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL] (top), and five classic methods (bottom) on binary trees.

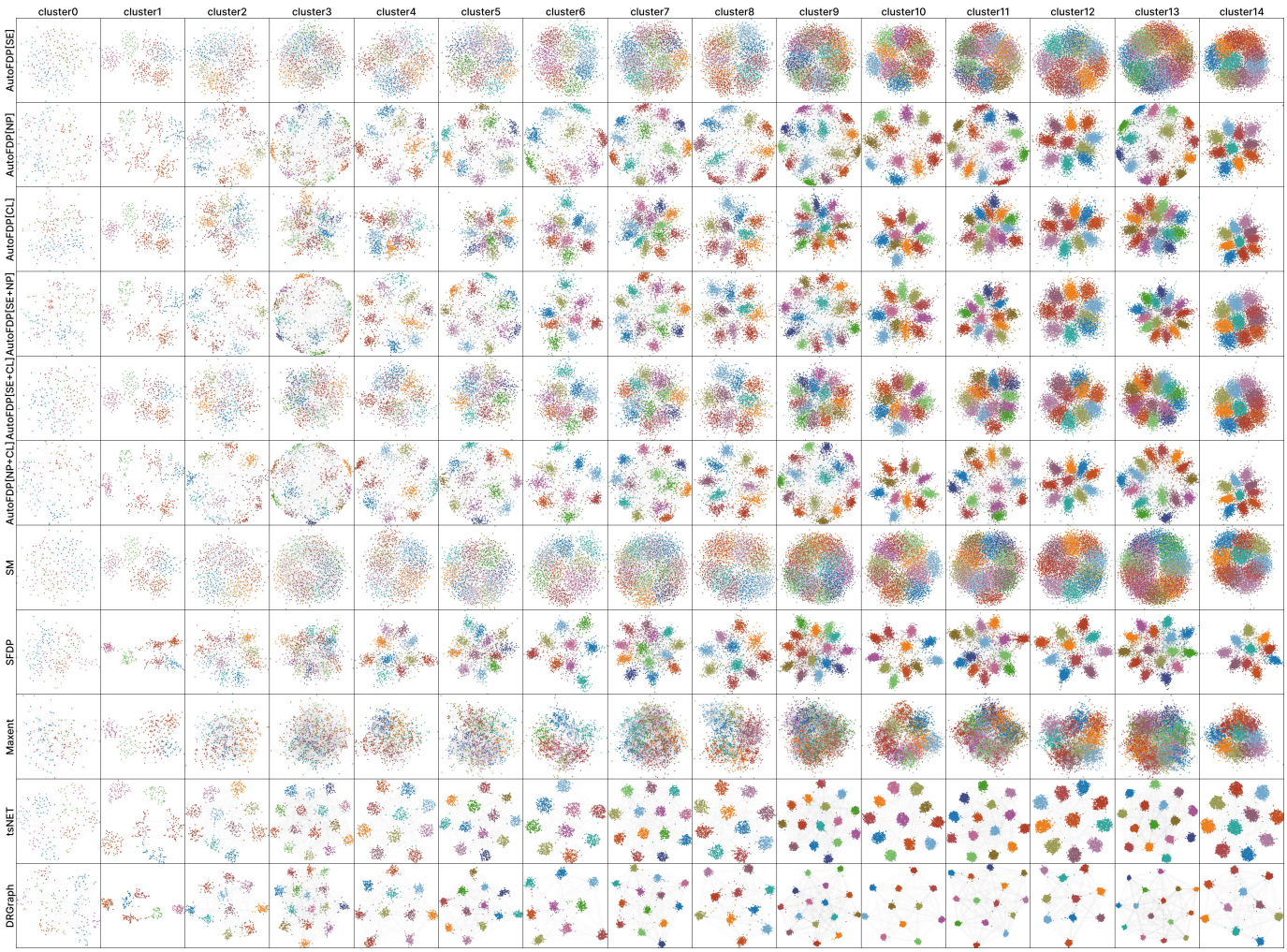

Fig. 12: Visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL] (top), and five classic methods (bottom) on clustered graphs.

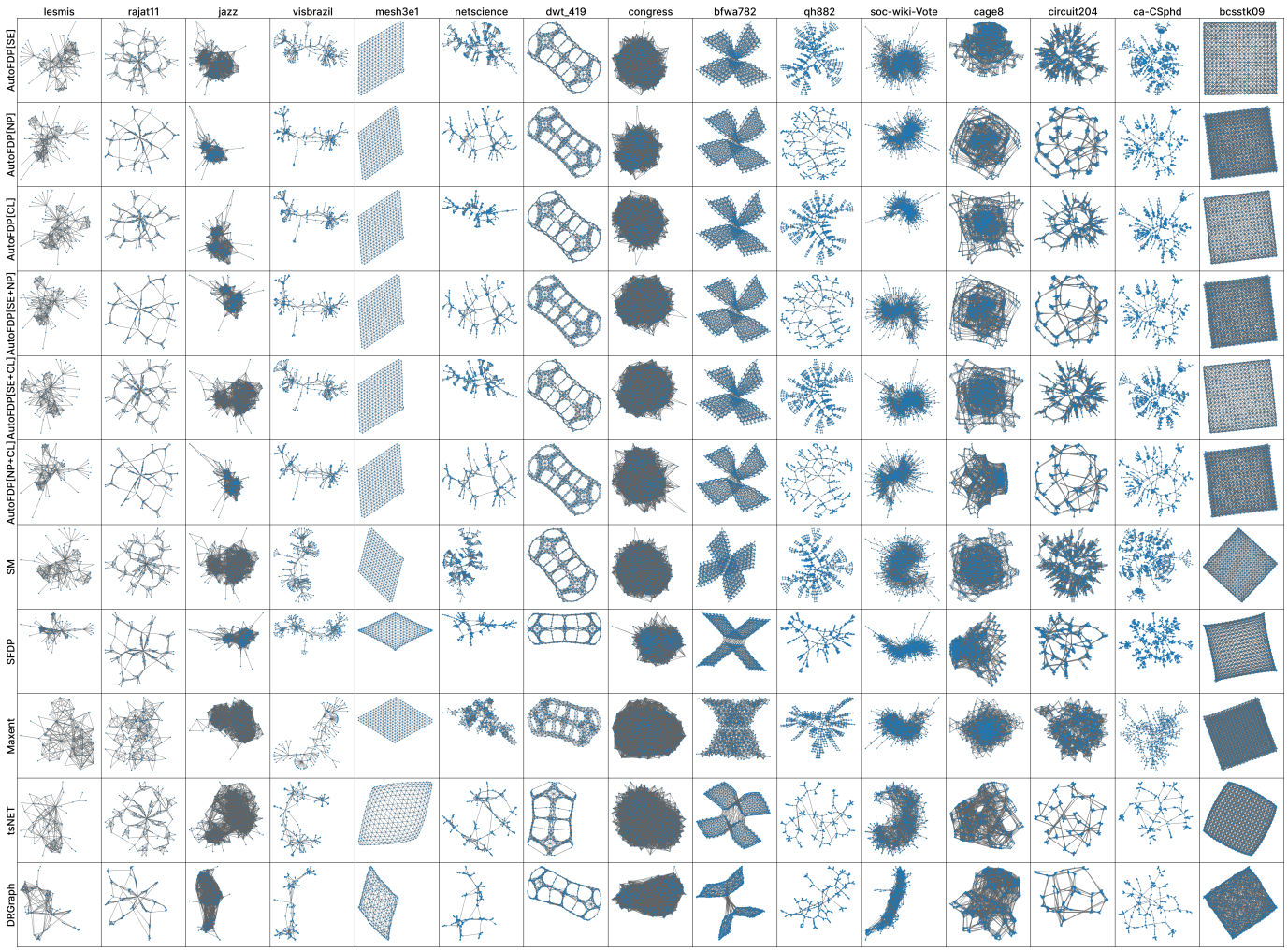

Fig. 13: Visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL] (top), and five classic methods (bottom) on the first 15 real graphs.

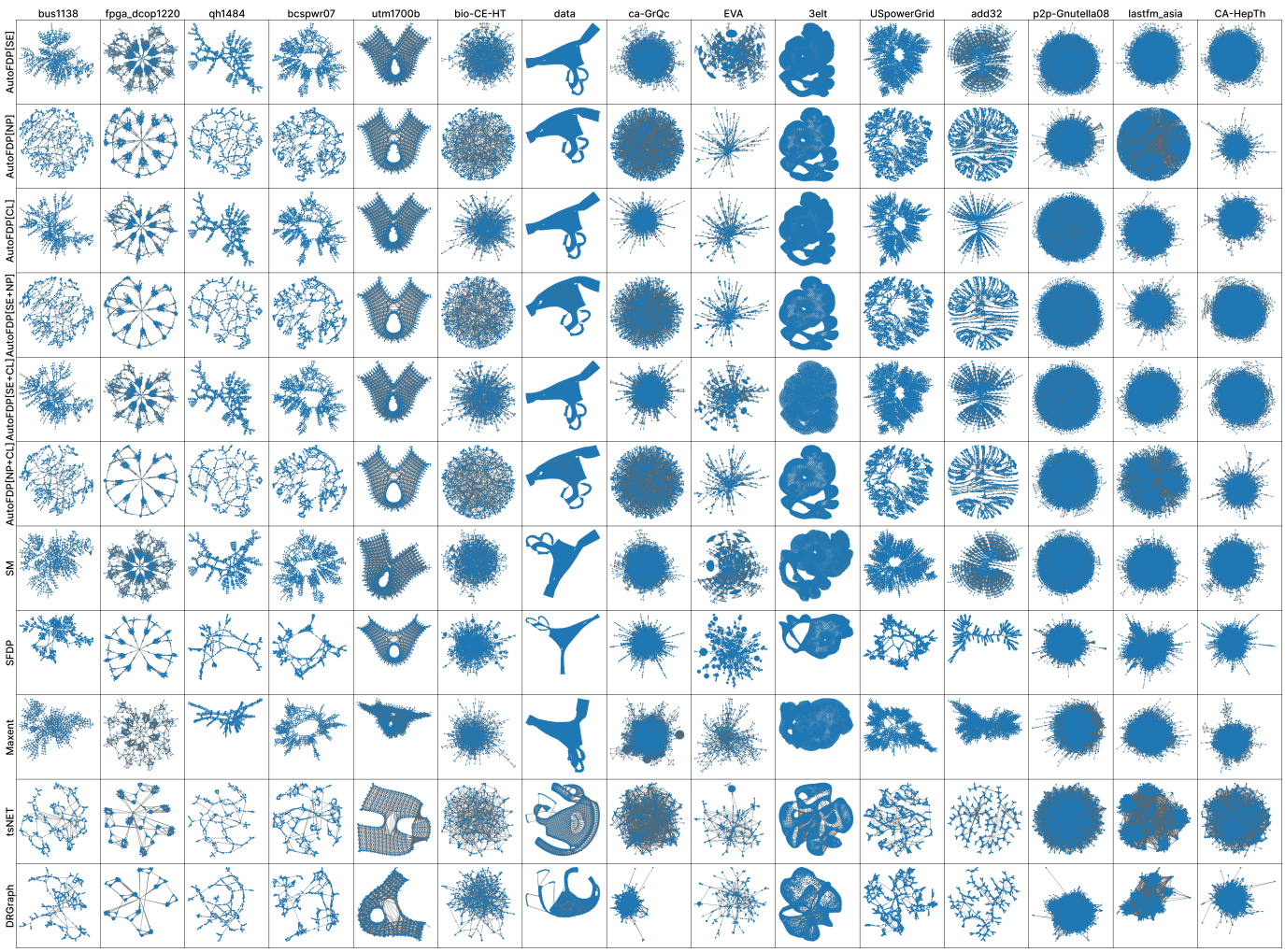

Fig. 14: Visual results obtained by using AutoFDP[SE], AutoFDP[NP], AutoFDP[CL], AutoFDP[SE+NP], AutoFDP[SE+CL], AutoFDP[NP+CL] (top), and five classic methods (bottom) on the last 15 real graphs.

4 Comparison to Deep Learning-based Methods

Last, we compared our method with the deep learning-based methods, DeepGD and SmartGD. Figure 15 and Figure 16 show the visualization results obtained by AutoFDP and DeepGD on 1000 graph data and 150 synthetic graphs selected from the Rome dataset, respectively. Similarly, Figure 19 and Figure 20 illustrate the results obtained by AutoFDP and SmartGD on the same subsets of the Rome dataset and the synthetic graphs, respectively. Due to the large number of visualizations, we show only 10 examples with different structures in the Rome dataset. For the synthetic graphs, we chose 12 different sizes of graphs to show. These include 4 clustered graphs, 4 binary tree graphs and 4 grid graphs. In Figure 17, Figure 18, Figure 21, and Figure 22 we show the visual results obtained by SmartGD and DeepGD on 27 real graphs, along with comparisons with the AutoFDP layout results. The 3 largest graphs p2p-Gnutella08, lastfm_asia and CA-HepThy were not included in the comparison because they consumed too much memory to run the results in DeepGD and SmartGD, but the corresponding results for AutoFDP can be found in section 3. In addition, Figure 23 shows the visual results of AutoFDP on optimizing shape-based metric on real graphs and the corresponding shape graphs, which are generated using the Relative Neighborhood Graph (RNG) method consistent with SmartGD.

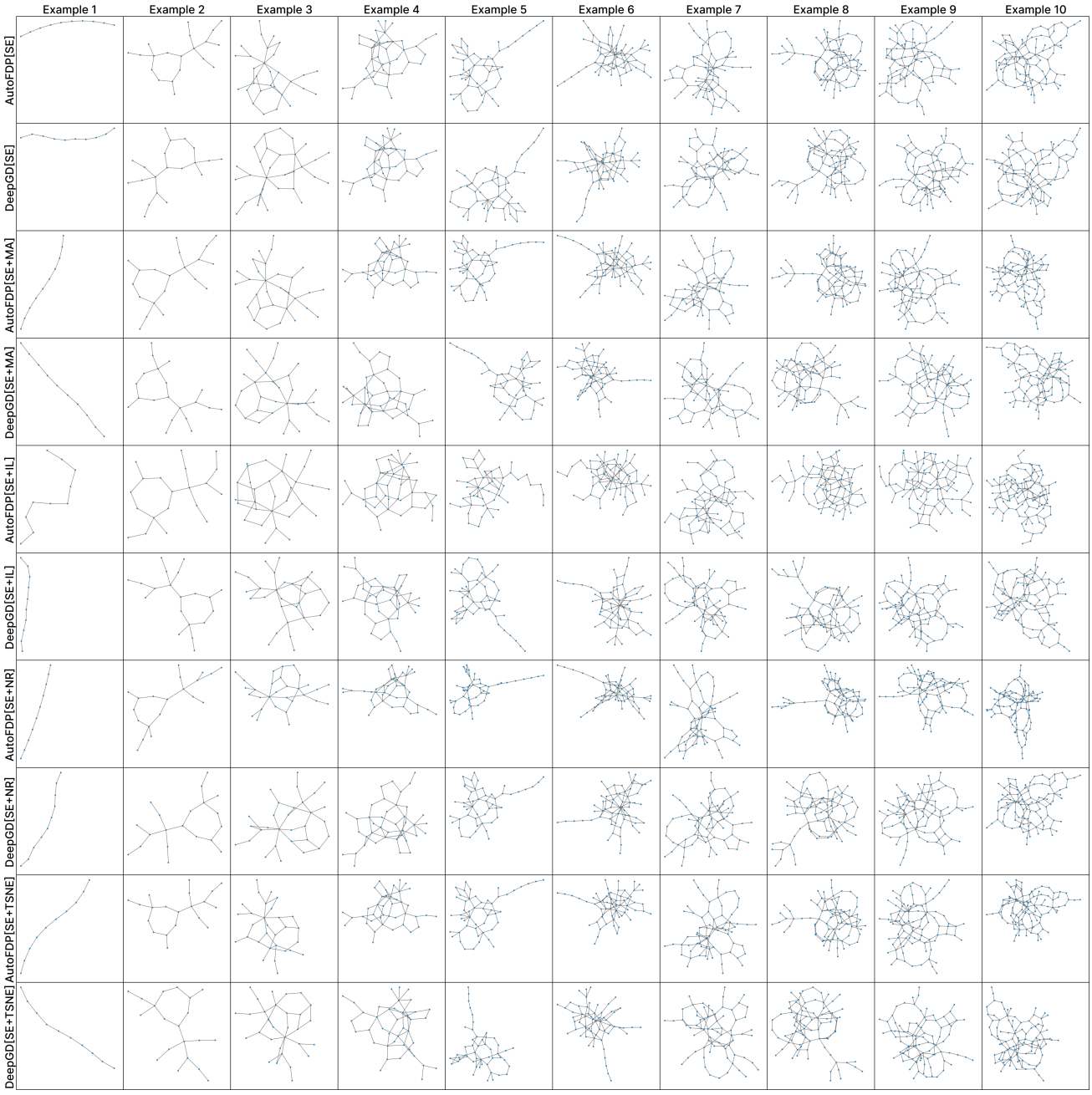

Fig. 15: Visual results generated by AutoFDP and DeepGD on Rome Dataset.

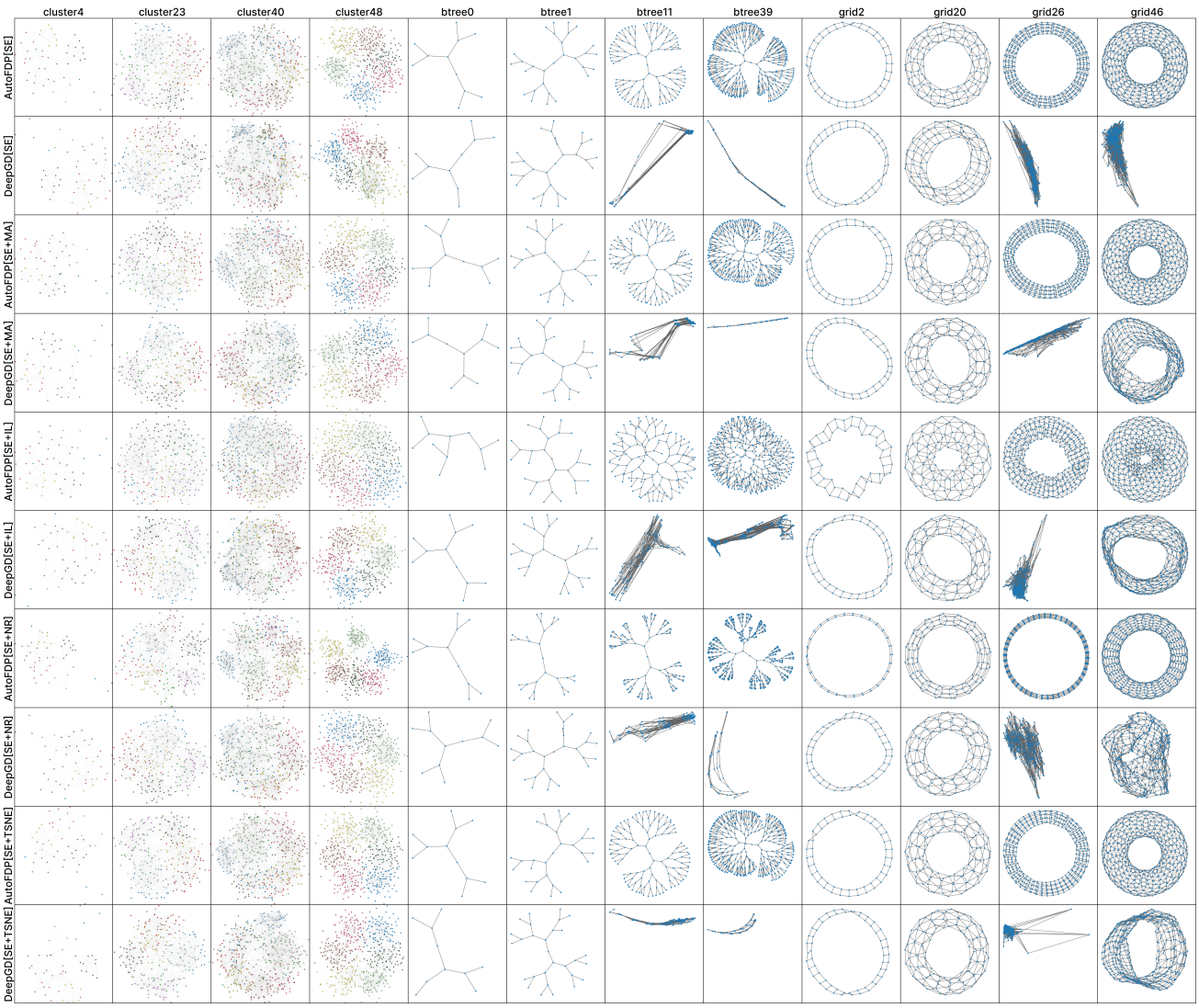

Fig. 16: Visual results generated by AutoFDP and DeepGD on synthetic graphs.

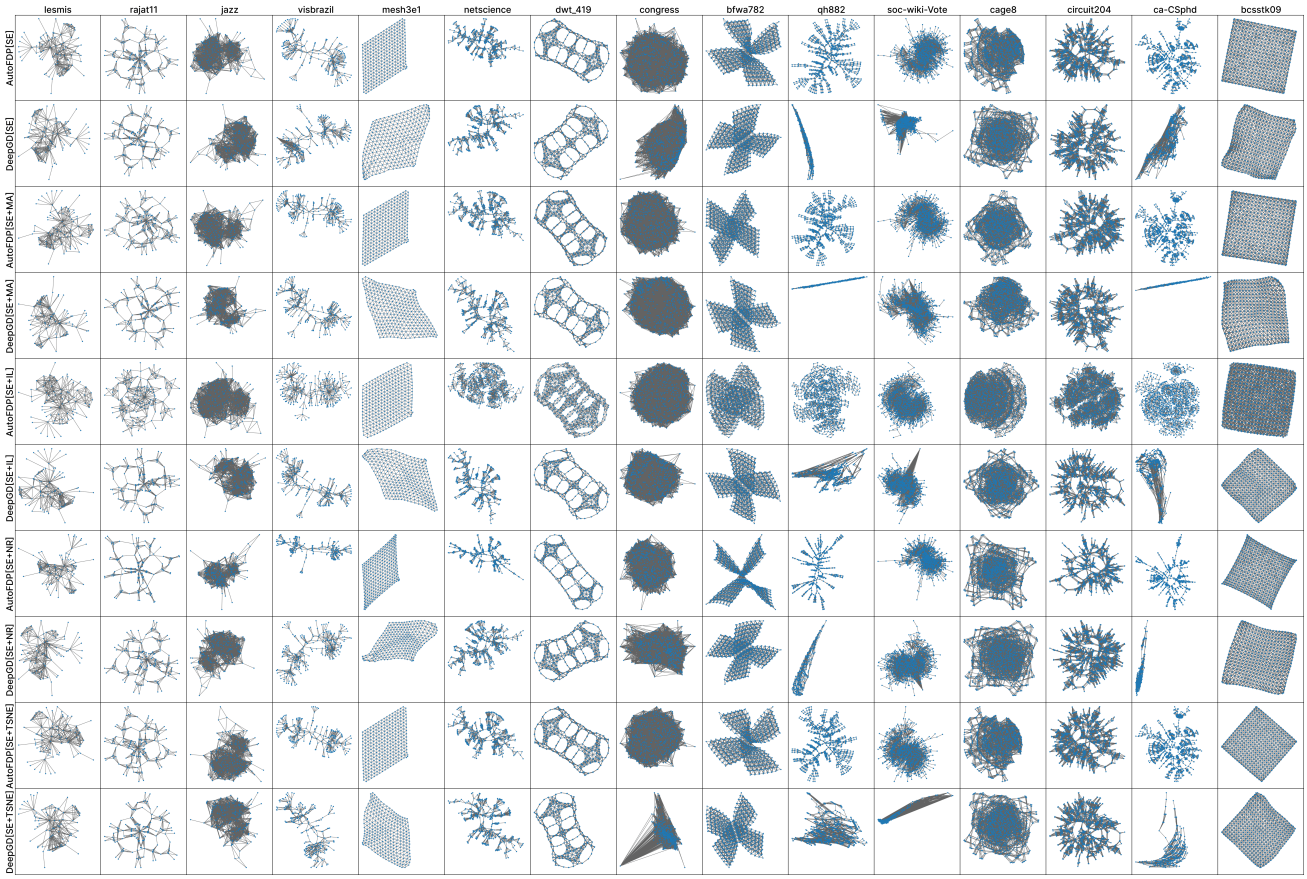

Fig. 17: Visual results generated by AutoFDP and DeepGD on the first 15 real graphs.

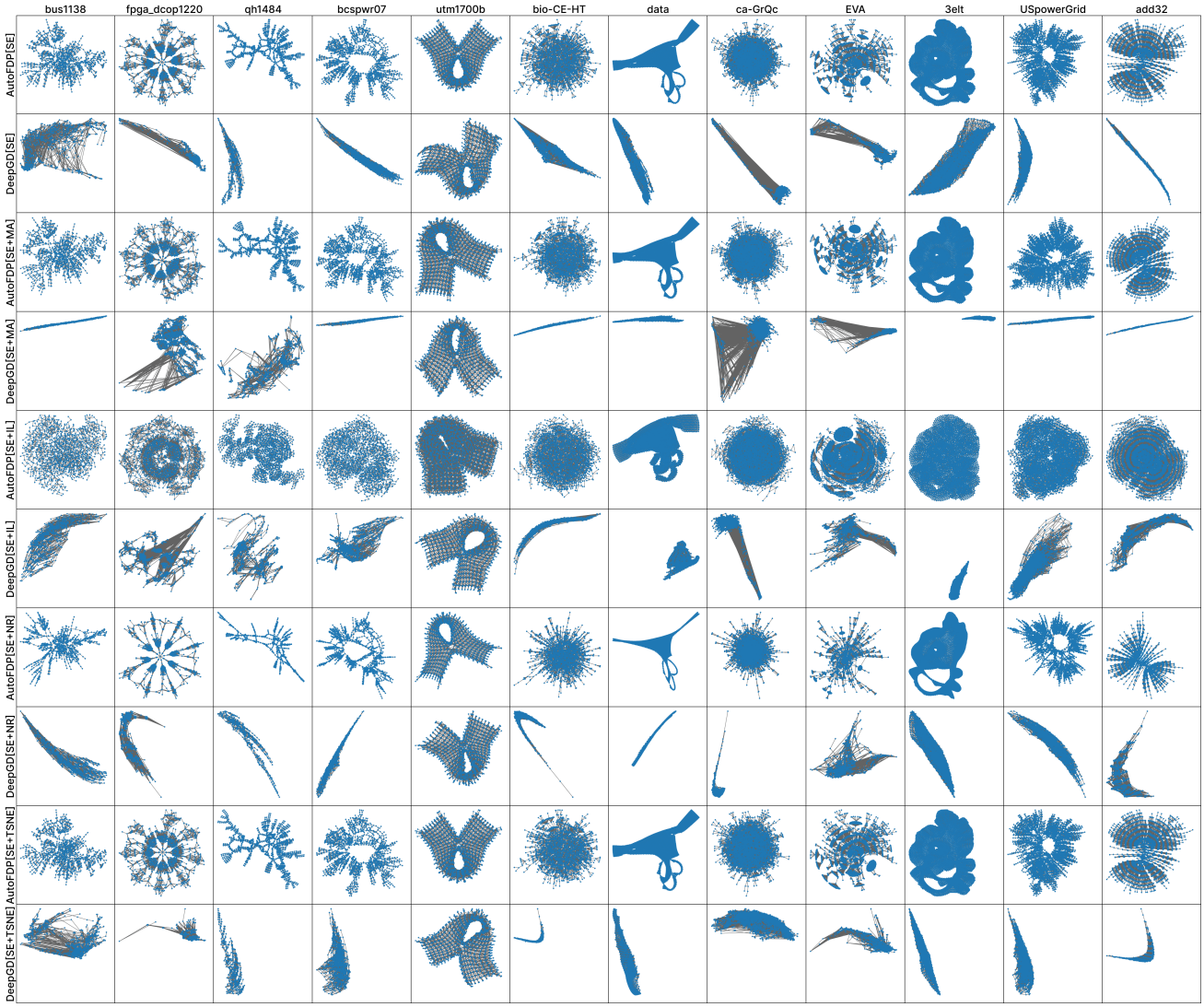

Fig. 18: Visual results generated by AutoFDP and DeepGD on the last 12 real graphs.

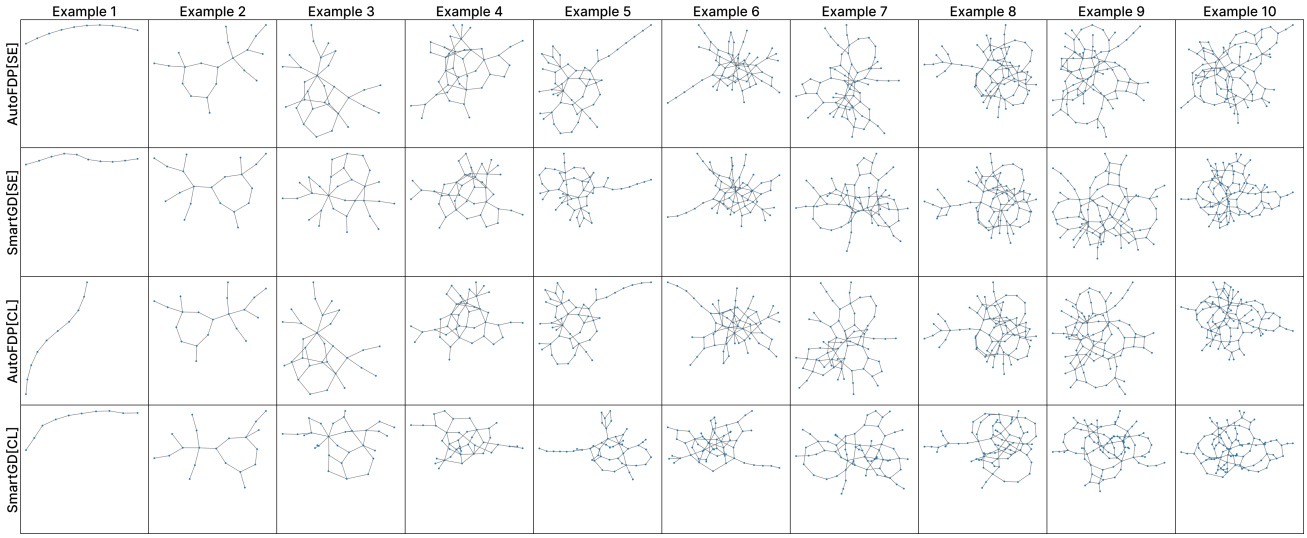

Fig. 19: Visual results generated by AutoFDP and SmartGD on Rome Dataset.

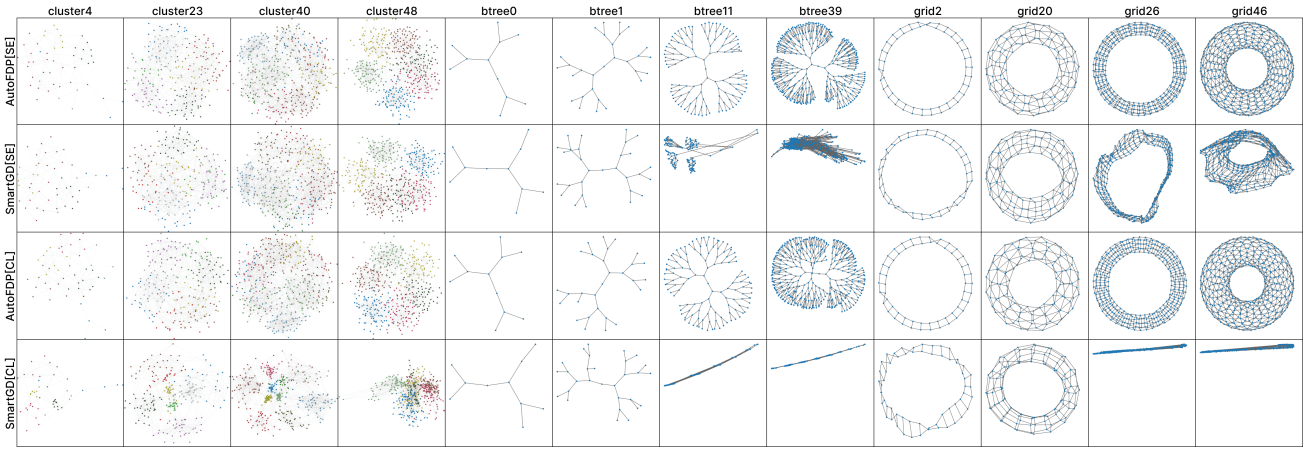

Fig. 20: Visual results generated by AutoFDP and SmartGD on synthetic graphs.

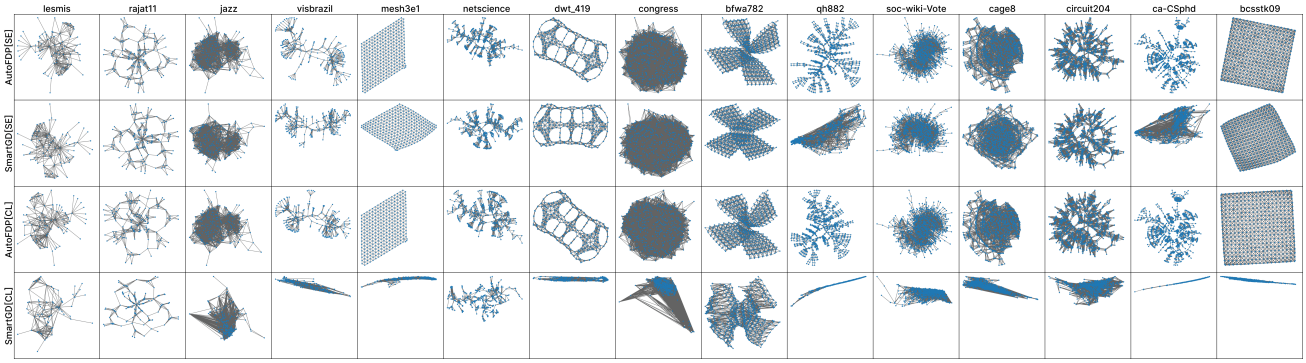

Fig. 21: Visual results generated by AutoFDP and SmartGD on the first 15 real graphs.

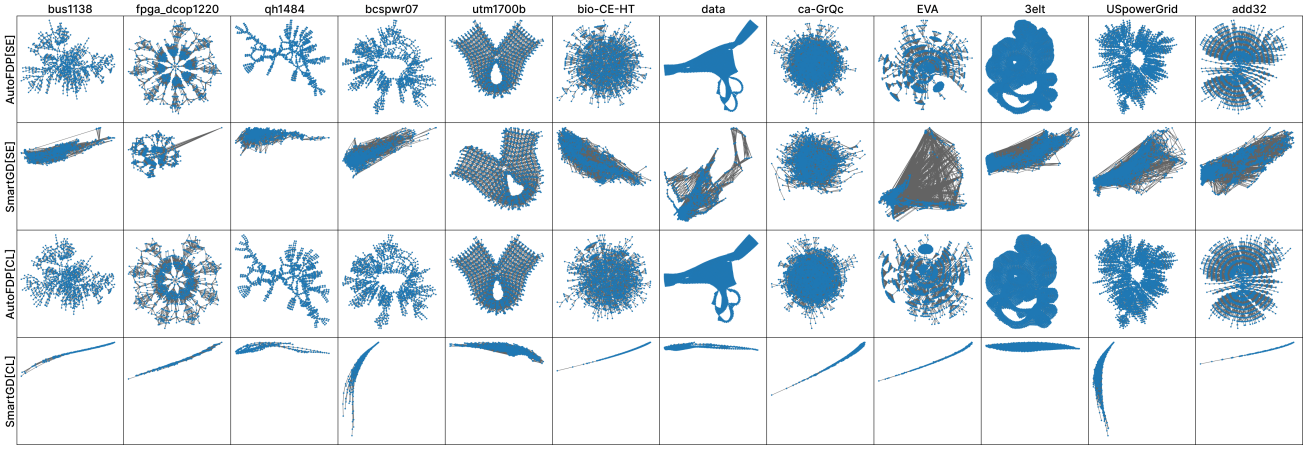

Fig. 22: Visual results generated by AutoFDP and SmartGD on the last 12 real graphs.

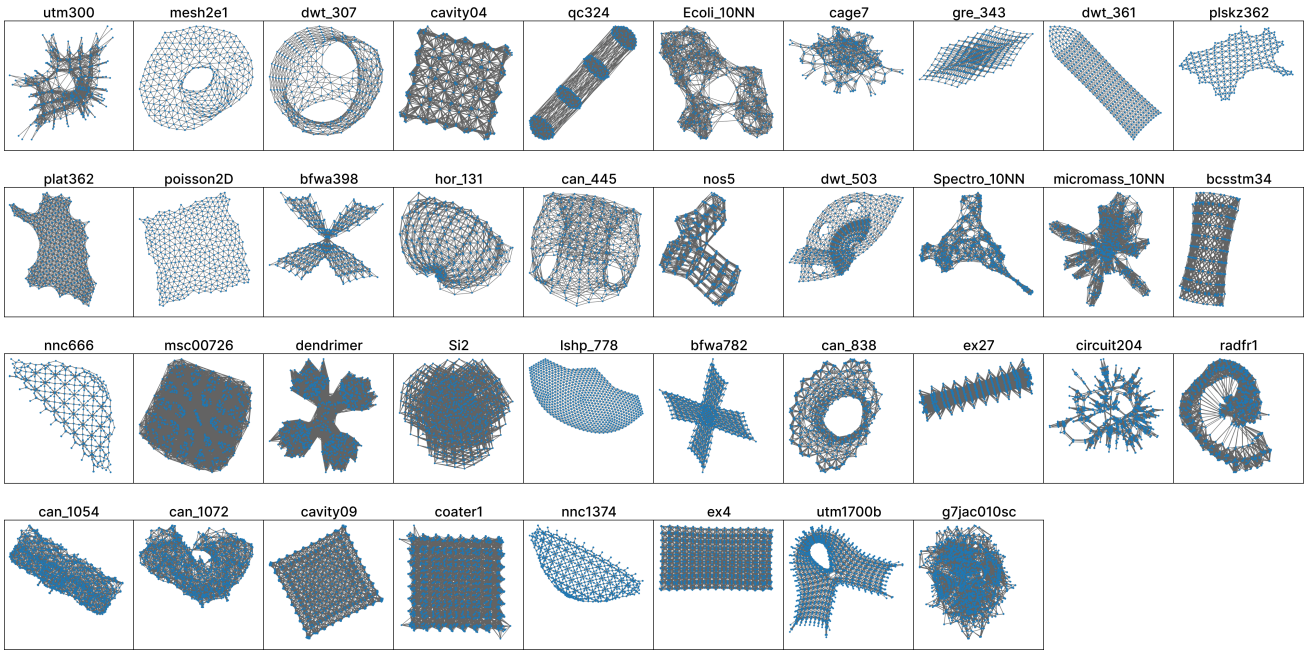

Fig. 23: Visual results obtained by AutoPDF optimizing shape-based criterion with corresponding RNG graphs.

REFERENCES

[1] R. Ahmed, F. De Luca, S. Devkota, S. Kobourov, and M. Li, "Multicriteria scalable graph drawing via stochastic gradient descent, $(SGD)^2$," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2388–2399, 2022. 1, 3

[2] X. Wang, K. Yen, Y. Hu, and H.-W. Shen, "DeepGD: A deep learning framework for graph drawing using GNN," *IEEE Computer Graphics and Applications*, vol. 41, no. 5, pp. 32–44, 2021. 1

[3] X. Wang, K. Yen, Y. Hu, and H. Shen, "SmartGD: A GAN-based graph drawing framework for diverse aesthetic goals," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 8, pp. 5666–5678, 2024. 1

[4] 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*, vol. 27, no. 2, pp. 1666–1676, 2020. 2, 11

[5] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea, "Graph layouts by t-SNE," *Computer Graphics Forum*, vol. 36, no. 3, pp. 283–294, 2017. 2, 11

[6] S. Di Bartolomeo, E. Puerta, C. Wilson, T. Crnovrsanin, and C. Dunne, "A collection of benchmark datasets for evaluating graph layout algorithms," Under submission to Graph Drawing Posters, 2023. [Online]. Available: https://visdunneright.github.io/gd_benchmark_sets/ 2

[7] A. Schulz, "Drawing 3-polytopes with good vertex resolution." *Journal of Graph Algorithms and Applications*, vol. 15, no. 1, pp. 33–52, 2011. 2

[8] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, pp. 4292–4293, 2015. [Online]. Available: https://networkrepository.com 2

[9] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *International Symposium on Graph Drawing*, 2004, pp. 239–250. 2

[10] U. Brandes and C. Pich, "Eigensolver methods for progressive multidimensional scaling of large data," in *Graph Drawing*, M. Kaufmann and D. Wagner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 42–53. 2

[11] E. R. Gansner, Y. Hu, and S. North, "A maxent-stress model for graph layout," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 6, pp. 927–940, 2012. 1, 2, 11

[12] Y. Hu, "Efficient high-quality force-directed graph drawing," *Mathematica Journal*, vol. 10, no. 1, pp. 37–71, 2005. 2, 11

[13] C. Walshaw, "A multilevel algorithm for force-directed graph drawing," in *Graph Drawing: 8th International Symposium, GD 2000 Colonial Williamsburg, VA, USA, September 20–23, 2000 Proceedings 8*. Springer, 2001, pp. 171–182. 2

[14] F. Grötschla, J. Mathys, R. Veres, and R. Wattenhofer, "CoRe-GD: A hierarchical framework for scalable graph visualization with GNNs," in *The Twelfth International Conference on Learning Representations (ICLR 2024)*. OpenReview, 2024. 2

[15] F. Zhong, M. Xue, J. Zhang, F. Zhang, R. Ban, O. Deussen, and Y. Wang, "Force-directed graph layouts revisited: a new force based on the t-distribution," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 7, pp. 3650–3663, 2024. 2

[16] H. C. Purchase, "Metrics for graph drawing aesthetics," *Journal of Visual Languages and Computing*, vol. 13, no. 5, pp. 501–516, 2002. 3

[17] M. Chrobak, M. T. Goodrich, and R. Tamassia, "Convex drawings of graphs in two and three dimensions (preliminary version)," in *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, 1996, pp. 319–328. 3

[18] P. Eades, S.-H. Hong, K. Klein, and A. Nguyen, "Shape-based quality metrics for large graph visualization," in *Graph Drawing and Network Visualization*. Springer International Publishing, 2015, pp. 502–514. 3

[19] J. X. Zheng, S. Pawar, and D. F. Goodman, "Graph drawing by stochastic gradient descent," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 9, pp. 2738–2748, 2018. 11