Challenges of Generating Structurally Diverse Graphs

Fedor Velikonivtsev
HSE University, Yandex Research
fvelikon@yandex-team.ru

Mikhail Mironov
Yandex Research
mironov.m.k@gmail.com

Liudmila Prokhorenkova
Yandex Research
ostroumova-la@yandex-team.ru

Abstract

We address the problem of generating graphs that are structurally diverse. First, we discuss how to define diversity for a set of graphs, why this task is non-trivial, and how one can choose a proper diversity measure. Then, for a given diversity measure, we propose and compare several algorithms optimizing it: we consider approaches based on standard random graph models, local graph optimization, genetic algorithms, and neural generative models. We show that it is possible to significantly improve diversity over basic random graph generators and generate graphs with diverse structural characteristics. This paper is an extended abstract that briefly describes the main ideas of our work, see [1] for the full version.

1 Introduction

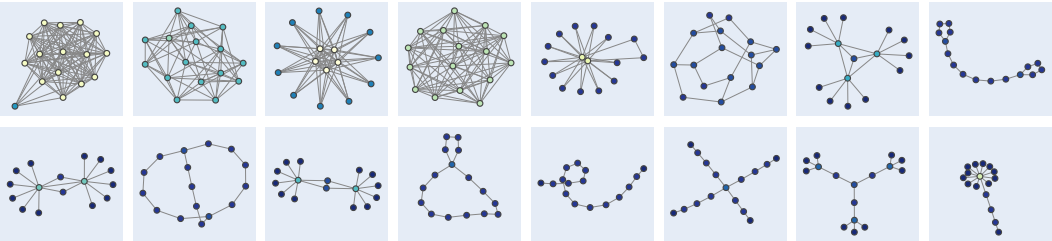

Figure 1: A sample of generated graphs

Many real-world objects can be naturally represented as graphs: biological and chemical entities, interaction networks, road maps, and so on. Hence, the analysis of graph-structured data is an important and rapidly developing research area. To generate realistic graph structures, many random graph models have been proposed [2]. Such models aim to imitate properties typically observed in natural structures: power-law degree distribution, small diameter, community structure, and others.

On the other hand, for some applications, it is important to have a set of graphs that are *structurally diverse*. For instance, if one needs to automatically verify the correctness of a graph algorithm, estimate how well a heuristic approach approximates the true solution for a graph problem, or evaluate neural approximations of graph algorithms [3]. In all these cases, algorithms and models should be tested on as diverse graph instances as possible since otherwise the results can be biased towards particular properties of the test set. In other words, we need representative graphs that 'cover' (in some sense) the space of all graphs.

To the best of our knowledge, the problem of generating a dataset where graphs are maximally diverse has not been addressed in the literature yet. In this paper, we fill this gap. For this purpose, we first need to define diversity of a set of graphs. This is already a challenging task and we discuss it in Section 2. After we have defined a performance measure for our problem, several approaches can be used to optimize it. We develop and analyze the following strategies: a greedy method based on diverse random graph generators, a local graph optimization approach, an adaptation of the genetic algorithm to our problem, and a method based on neural generative modeling.

Y. Zhu et al., Challenges of Generating Structurally Diverse Graphs (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

We empirically investigate the proposed strategies and show that they indeed allow us to generate structurally different graphs. In particular, we analyze various graph structural characteristics and show increased diversity of their joint distribution. Moreover, since we consider diversity measures based on several graph distances, our results shed light on the properties of these graph distances. Indeed, depending on the function we optimize, the structural properties of the generated graphs can vary since graph distances focus on different aspects of graph dissimilarity. Thus, by inspecting the properties of generated graphs, one can better understand what graph characteristics a particular graph distance is sensitive to.

2 Defining diversity for a set of graphs

This section discusses how to define diversity and why it is non-trivial. Intuitively, diverse graphs are expected to cover (in some sense) the space of all graphs.[1] However, just sampling graphs uniformly at random from the set of all graphs (i.e., using the Erdős-Rényi model with $p = 0.5$) would not give us diverse graphs. Indeed, it is known that with high probability graphs generated according to the Erdős-Rényi model have very similar properties [4].

Intuitively, by *diverse graphs* we mean those having different structural properties such as degree distribution, pairwise distances, subgraph counts, and so on. This intuition is hard to formalize as one may potentially come up with countless properties. Defining graph dissimilarity is closely related to *graph distances*. Graph distances have been studied for a long time, and many variants exist in the literature [5]. Each graph distance captures particular graph properties and our paper does not aim to answer which distance is better. In our experiments, we consider several representative options.

Now, assume that we have a multiset of N graphs $S = G_1, \dots, G_N$. Throughout the paper, we consider undirected graphs without self-loops and multiple edges. Assume that we are given a distance measure $D(G, G')$ that evaluates dissimilarity between two graphs. Then, we define diversity as:

$$\text{Diversity}(S) = F(\{D(G, G') : G, G' \in S\}), \tag{1}$$

where F is some function that computes diversity given a set of pairwise distances.

It remains to define the function F. Several variants have been considered in the literature [6, 7]. Some popular options are *average* and *minimum* pairwise distance between the objects. The shortcoming of *minimum* is that it is not sensitive to most of the distances (only to the smallest one). On the other hand, optimizing *average* may lead to degenerate undesirable solutions.[2]

Motivated by the shortcomings of existing measures, we propose an alternative one inspired by the *energy of a system of equally charged particles*. Namely, we define the *energy* of a set of graphs S as

$$-\frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{D(G_i, G_j)}. \tag{2}$$

This function can be naturally interpreted as the average pairwise energy for a system of equally charged particles (we multiply by -1 to get a measure that is larger for more diverse sets of graphs).

3 Algorithms for diversity optimization

After we have defined the distance $D(\cdot, \cdot)$ and the measure of diversity, our primary goal is to find a multiset of graphs $\bar{S}$ of size N to maximize its diversity:

$$\bar{S} = \underset{G_1, G_2, \dots, G_N \in \mathcal{G}_n}{\arg\max} \text{Diversity}(\{G_1, G_2, \dots, G_N\}), \tag{3}$$

where $\mathcal{G}_n$ is the set of all graphs with n nodes.

We aim to investigate diverse algorithms for diversity optimization: from a basic approach based on random graph generators to a more advanced one based on neural generative modeling. Note that the proposed algorithms can be applied to any given measure of diversity.

[1] In this work, we use the terms 'diversity' and 'coverage' interchangeably.

[2] We refer to the full version of the paper [1] for a detailed discussion of popular diversity measures, examples of their undesirable behavior, and a formal approach to analyzing measures of diversity.

Greedy algorithm. The main idea of this algorithm is to build a set of diverse graphs iteratively, by adding at each step the most suitable graph from a predefined set $\hat{S}$ of a much larger size. This set is not specified: it can be either user input, the result of another algorithm, or a set of graphs generated by random graph models. The process initiates by randomly choosing a graph from $\hat{S}$. At each step, we have a set of already chosen graphs S and add one more graph from $\hat{S}$ that gives the largest diversity improvement. In the full version [1], we provide more details, the analysis of the computational complexity of this algorithm, and a lower bound on the diversity of graphs returned by the greedy algorithm relative to the diversity of the initial set $\hat{S}$.

Genetic algorithm. The genetic algorithm enhances the diversity of a graph population through evolutionary operations. Starting with an initial set of N graphs, it iteratively refines this set by selecting pairs of graphs as parents and generating a child through crossover and mutation processes. This child can replace one of the graphs in the population if it increases the overall diversity; otherwise, the algorithm tries to find more suitable offspring by repeating the process. To prevent itself from getting stuck in local optima, the algorithm can accept a candidate that decreases the overall diversity if the number of unsuccessful attempts exceeds a certain threshold. The algorithm iterates for a predefined number of iterations, ultimately evolving the population towards greater diversity. This approach adapts principles from genetics to solve optimization problems, as we try to preserve beneficial graph characteristics, at the same time introducing novel configurations to achieve a diverse set of graphs, see [1] for more details.

Local optimization algorithm. The main idea of the local optimization algorithm is the refinement of the diversity of a graph population by iteratively modifying individual graphs. Starting from an initial set, we randomly sample graphs and make small modifications to their structure (single edge addition/deletion). Then, if the overall diversity improves, we accept the change. As in the other algorithms, we can accept unsuccessful modifications after consecutive failed attempts to prevent stagnation at a local optimum. Since local optimization makes small modifications at each step, this approach is expected to be most efficient when the input set of graphs is already sufficiently diverse. Thus, when we combine several algorithms, local optimization is always the last step.

Iterative graph generative modeling. Neural generative models are known to be a powerful tool for generating graphs that imitate a given distribution [8–10]. Hence, we aimed to investigate whether such approaches can be used for generating graphs that are structurally diverse. However, for this task, there is no predefined distribution that needs to be captured. We address this via the following iterative procedure. The process starts from an initial graph set S_0 and then iteratively enhances the diversity. At each iteration, the current set of graphs S_i is used to train a generative model. Then, this model is used to generate a significantly larger set of new graphs. From this new set, a smaller subset of diverse graphs S_{i+1} is selected via the greedy approach. We expect that S_{i+1} is more diverse than S_i. So, we repeat the process by training a neural generative model on the new set S_{i+1}. For the neural network architecture, we use Discrete Denoising Diffusion Model (DiGress) [10]. We refer to the full paper [1] for a more detailed description of this approach.

4 Experiments

Setup. In our experiments, we consider four representative distance measures: heat and wave NetLSD [11], Graphlet Correlation Distance [12], and Portrait Divergence [13]. We select these distances to be diverse: NetLSD is based on the Laplacian eigenvalues (we use NetLSD-heat and NetLSD-wave kernels), Graphlet Correlation Distance (GCD) uses local structures, while Portrait Divergence (Portrait-div) takes into account both local and global properties.

We evaluate the following approaches described in Section 3: Greedy, Genetic, Local Optimization (LocalOpt), and iterative graph generative modeling (IGGM). Our evaluation also includes the comparisons against simple baseline models, specifically the Erdős-Rényi graphs sampled with various p (ER-mix) and a sample from diverse random graph generators (see [1]). In most of the experiments, we generate $N = 100$ graphs with $n = 16$ vertices. We also conduct experiments with non-neural algorithms on the set of 100 graphs with size $n = 64$.

Examples of generated graphs. Since in our main experiments we generated 100 graphs, each having only 16 nodes, it is possible to visually inspect the generated graphs. To illustrate that the

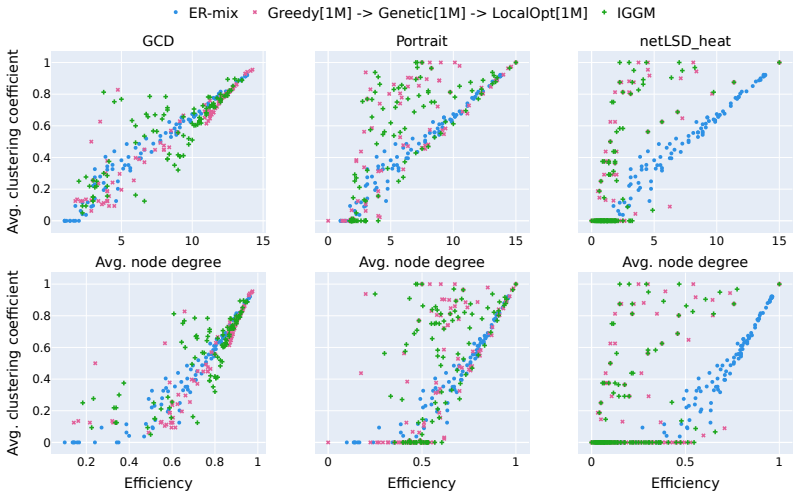

Figure 2: Joint distribution of graph characteristics for GCD, Portrait-div, NetLSD-heat

generated graphs have very different structural patterns, we show some examples in Figure 1. This sample of graphs is chosen from the resulting set of the Genetic algorithm with diversity based on Portrait-div. It is clear that graphs vary in density, internal structure, number of cycles, and planarity. Importantly, these graphs are clearly distinct from the input distribution ER-mix. More examples showing all generated graphs are shown in Figures 3-7. We see that when combined with Portrait-div, both Genetic and IGGM generate visually diverse and interesting structures. One can also notice that NetLSD tends to generate many extremely sparse graphs, while GCD generates more dense graphs.

Analysis of structural characteristics. Additionally, we analyze the structural characteristics of generated graphs. Figure 2 visualizes various characteristics for the ER-mix baseline, IGGM, and the combination of Greedy, Genetic, and LocalOpt. Obtaining a set of graphs in which an individual characteristic is diverse is easy: this can be achieved with the basic ER-mix. Hence, we visualize the joint distributions of pairs of characteristics.

It is clearly seen that compared to ER-mix, our algorithms lead to significantly more diverse pairs of characteristics. Also, it is worth mentioning that we often should not expect to cover all possible combinations: for instance, if the average degree is close to its maximal achievable value $n-1$, then the clustering coefficient has to be close to 1.

Visualizing pairwise graph characteristics can also help in the analysis and comparison of different graph distances. As can be seen in Figure 2, the generated graphs significantly depend on a particular graph distance used for computing diversity, see the full paper [1] for a detailed discussion.

In [1], we also conduct extensive numerical analysis of all the proposed algorithms and their various combinations. We conclude that all the algorithms significantly improve the performance over simple graph generators (see Table 1 in Appendix). Among the non-neural approaches, the best performance is achieved by a combination of Greedy, Genetic, and LocalOpt (applied in this order). In turn, the neural-network-based method IGGM gives a significant boost in diversity for GCD and Portrait-div distances and exhibits comparative results for NetLSD-heat. However, IGGM is more computationally expensive compared to other approaches. We refer to [1] for a detailed discussion.

5 Conclusion

In this work, we formulate the problem of generating structurally diverse graphs, define what it means for a set of graphs to be diverse, and propose various approaches to address this problem. Via a series of experiments, we show that the proposed approaches are capable of generating diverse graphs, both in terms of diversity measures and structural characteristics. We hope that our work will encourage researchers to dive deeper into this research direction. One particularly important and challenging direction for future research is scalability of the approaches to singificantly larger graphs.

References

[1] Fedor Velikonivtsev, Mikhail Mironov, and Liudmila Prokhorenkova. Challenges of generating structurally diverse graphs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2, 3, 4

[2] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006. 1

[3] Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021. 1

[4] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5(1):17–60, 1960. 2

[5] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing methods for comparing networks. *Scientific Reports*, 9(1):1–19, 2019. 2

[6] Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. 2

[7] Yutong Xie, Ziqiao Xu, Jiaqi Ma, and Qiaozhu Mei. How much space has been explored? Measuring the chemical space covered by databases and machine-generated molecules. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[8] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR, 2018. 3

[9] Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators, 2022.

[10] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[11] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. NetLSD: hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2347–2356, 2018. 3

[12] Ömer Nebil Yaveroğlu, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. Revealing the hidden language of complex networks. *Scientific Reports*, 4:4547, 2014. 3

[13] James P. Bagrow and Erik M. Bollt. An information-theoretic, all-scales approach to comparing networks. *Applied Network Science*, 4(1), 2019. 3

Table 1: Energy optimization results; the smaller value indicates greater diversity

Setup	GCD	Portrait-div	NetLSD-heat	NetLSD-wave
ER-mix	0.281	43.057	72.387	0.583
Random Graph Generators	0.553	6.009	116.685	1.334
Greedy[3M]	0.156	1.274	0.681	0.123
ER-mix→Genetic[3M]	0.139	1.264	0.677	**0.117**
Greedy[1M]→Genetic[2M]	0.139	1.263	0.674	0.118
ER-mix→Genetic[1M]→LocalOpt[2M]	0.138	1.259	0.675	**0.117**
Greedy[1M]→LocalOpt[2M]	0.139	1.255	0.679	0.118
Greedy[1M]→Genetic[1M]→LocalOpt[1M]	0.135	1.245	**0.673**	**0.117**
IGGM[1M]	**0.120**	**1.213**	0.675	0.148

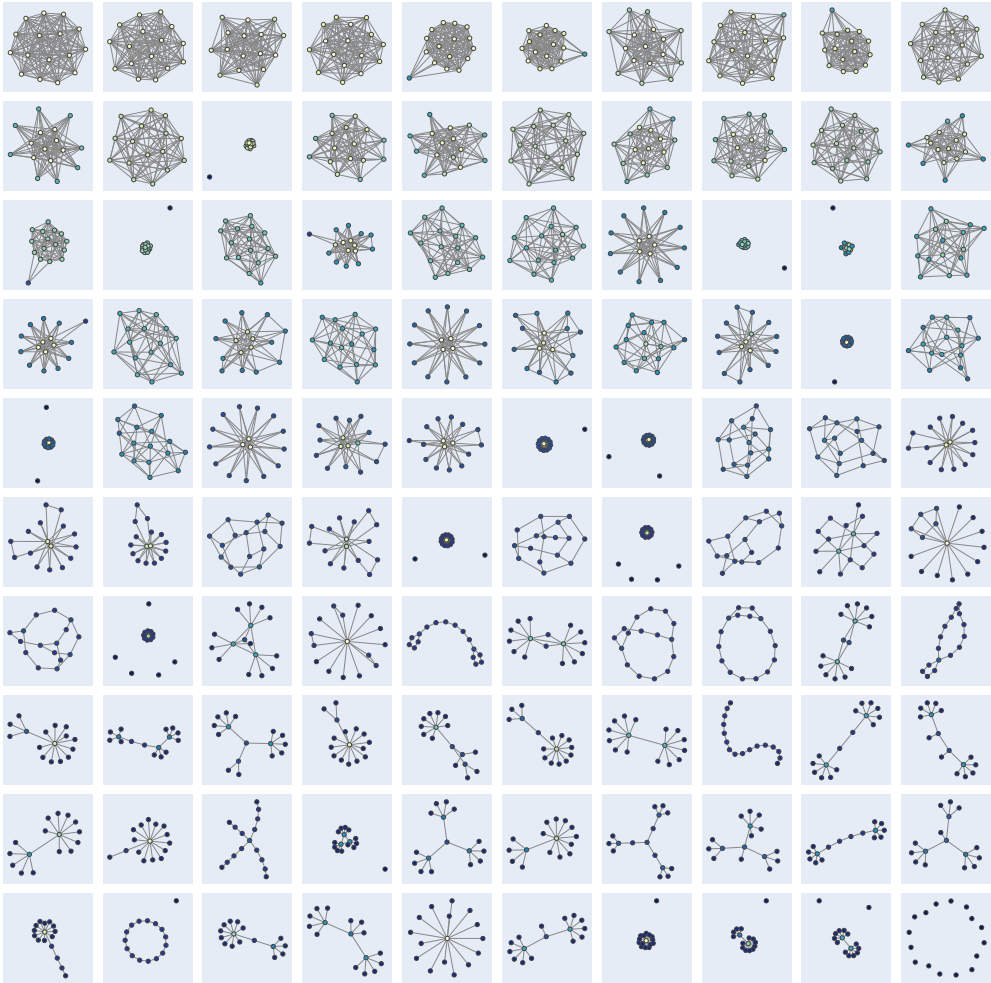

Figure 3: Graphs from Genetic with Portrait-div

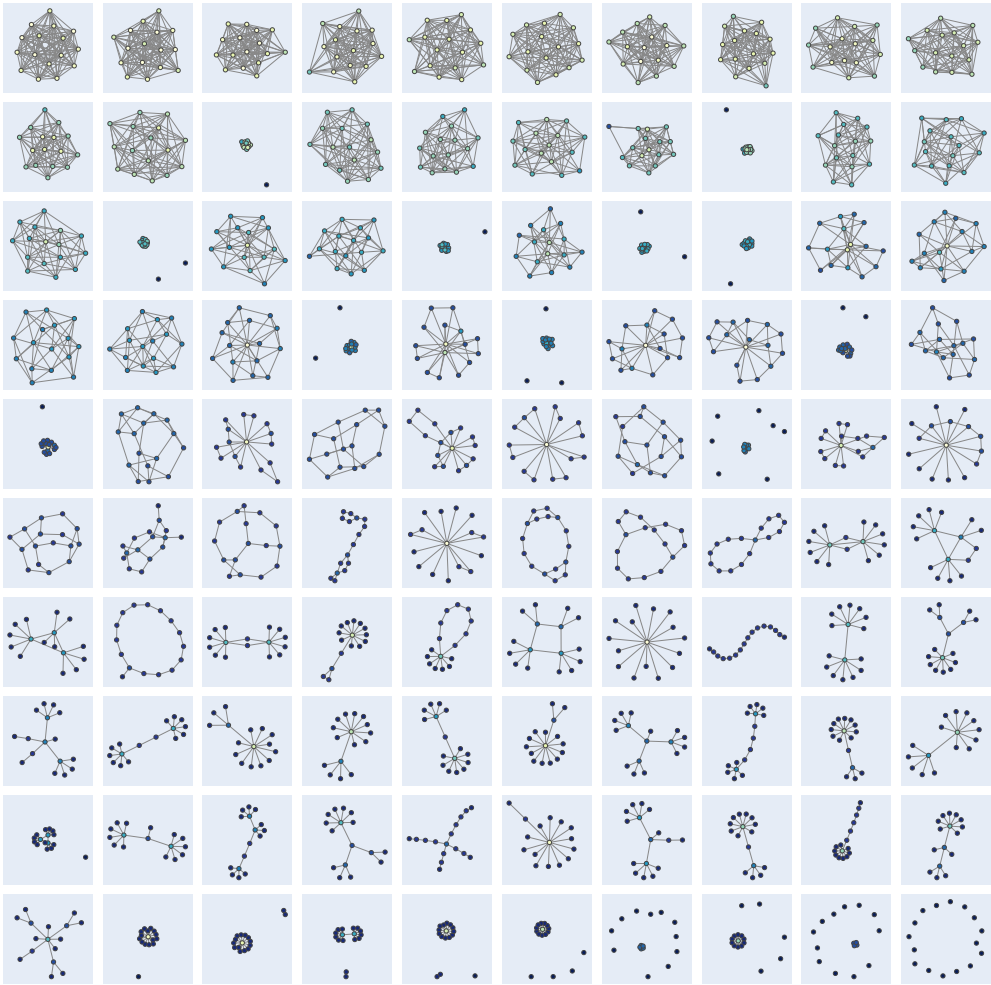

Figure 4: Graphs from IGGM with Portrait-div

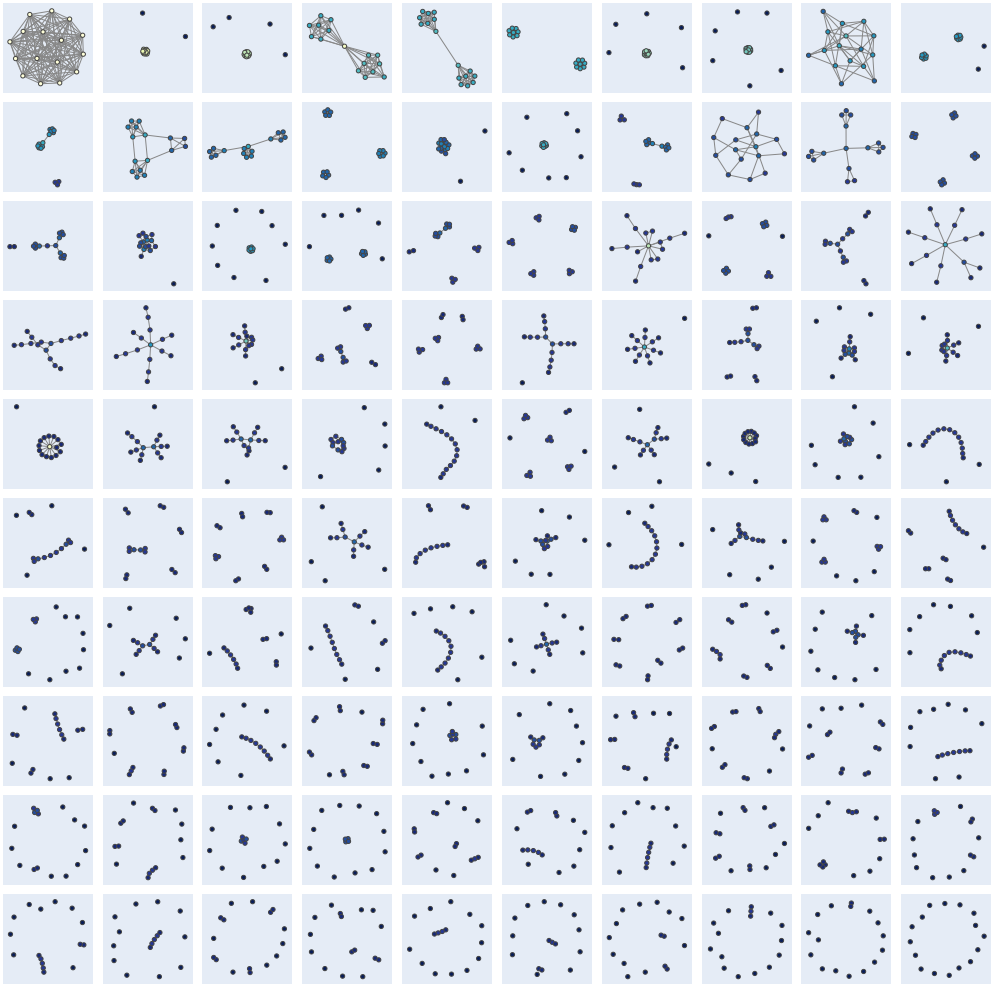

Figure 5: Graphs from IGGM with netLSD-heat: most of the graphs are sparse

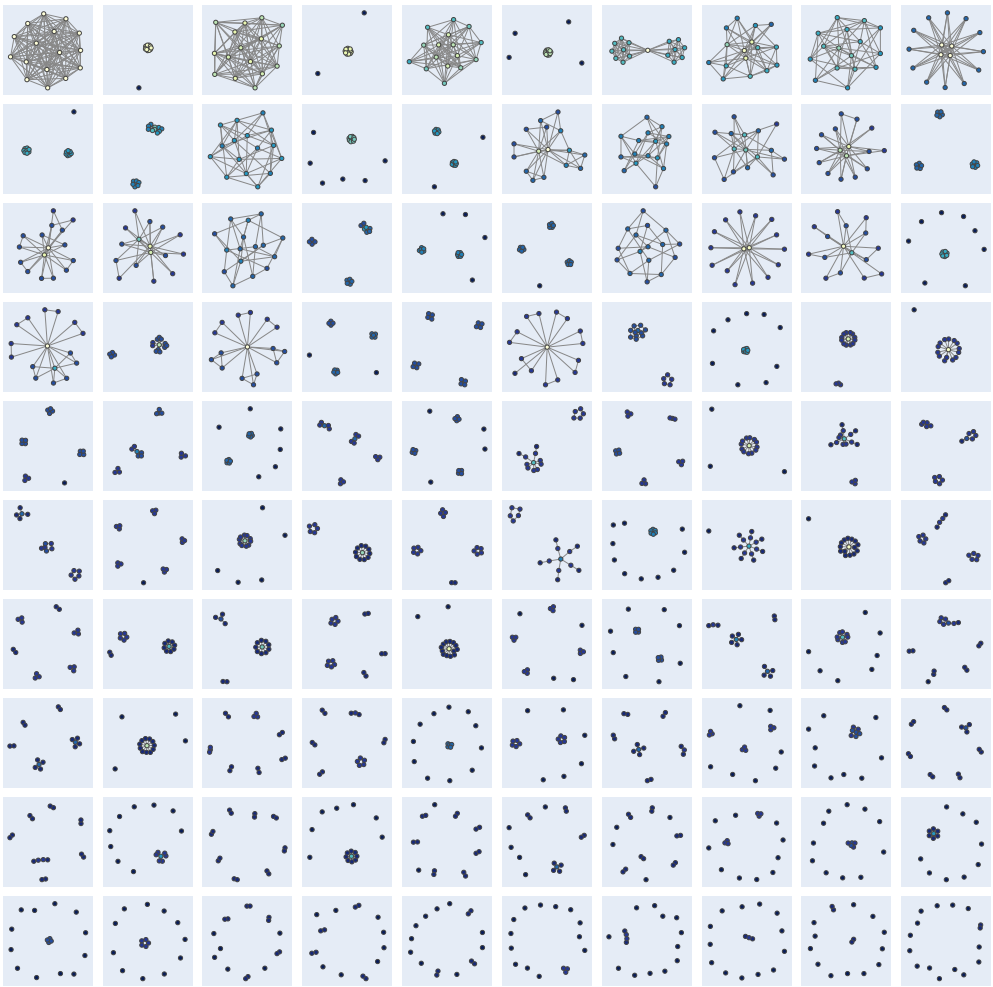

Figure 6: Graphs from IGGM with netLSD-wave: most of the graphs are sparse

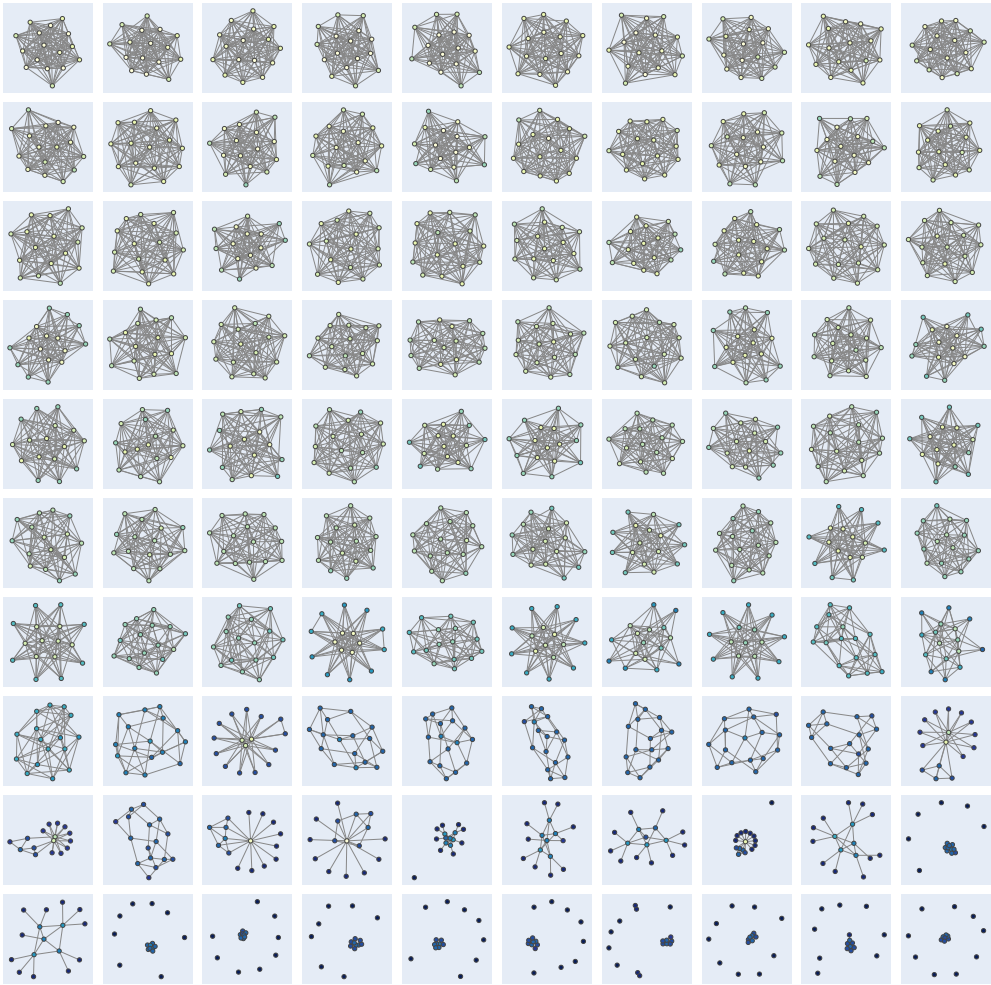

Figure 7: Graphs from IGGM with GCD