



## Structural bioinformatics

# Robustness of Deep Learning with Graph Convolutional Networks in Multi-Omics Data Integration for Small Clinical Samples

Giuliana Orizzonte<sup>1</sup>, Hae-Won Uh and Said el Bouhaddani

Julius Center, dept. Data Science and Biostatistics, University Medical Center Utrecht, Universiteitsweg 100, 3584 CG, Utrecht, The Netherlands

\*Corresponding author. g.orizzonte@uu.nl

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

### Abstract

**Motivation:** .

**Results:** .

**Availability and Implementation:** .

**Supplementary information:** The software is available at <https://github.com/gorizzonte>

### Introduction

Recent advancements in the life sciences resulted in an unprecedented abundance of biological data. The data, although comprehensive, is often characterized by small sample sizes, high dimensionality, and heterogeneous measurement techniques [1]. Even a single cell exhibits a myriad of molecular interactions and pathways, spanning layers of biological functions that include genomics, transcriptomics, proteomics, metabolomics, and more, all of which constantly interact [2, 3]. These diverse layers of molecular information are collectively referred to as multi-omics [4]. Each omics layer produces data of varying scales, formats, and dimensions, and the process of combining information is known as multi-omics data integration [5, 6, 7]. Multi-omics integration can help understand disease mechanisms, treatment responses, and individual variations at a molecular level [8]. By integrating genomics, transcriptomics, proteomics, metabolomics, etc. researchers can analyze the biological processes involved in diseases as a whole [9, 10]. For instance, by analyzing gene expression (transcriptomics) alongside protein activity (proteomics), scientists can trace how genetic alterations influence protein function and cellular behavior [11]. This approach can identify biomarkers for diagnosis, prognosis, and therapy development [12, 13, 14].

However, traditional analytical techniques have limitations that hinder such a global analysis; often they are limited to single omics analyses or restrict relations between omics to be linear [15, 16, 17]. Additionally, they overlook complex, non-linear interactions, which could render the results incomplete [18, 19, 20].

Deep-learning methods, particularly Neural Networks (NNs), have furthered multi-omics data analysis because they excel in identifying patterns and inferring relationships from large, multi-dimensional datasets [21, 22, 23]. For instance, they can recognize patterns in gene expression data that may be indicative of disease states or therapy responses [24, 25, 26]. Graph Neural Networks (GNNs) are a class of NN models that deal with graph-structured data [27]. The graph structure can reproduce existing biological networks, such as protein-protein interaction networks or gene regulatory networks [28]. This structural similarity enables GNNs to simultaneously capture biological entities (e.g., proteins, genes) as well as their interactions [29, 30, 31]. In such graphs, entities are represented as nodes while connections between nodes are represented as edges. GNNs also allow for message-passing algorithms, whereby nodes send and receive 'messages', i.e., information, to and from their neighbors [32, 33]. The shared information is used to update the features of each node.

The iterative process allows GNNs to refine the representation of each node based on its local neighborhood, capturing both individual characteristics and the broader context of its connections [34, 35]. This mechanism is useful in the analysis of biological systems because the functions and behaviors of an entity are determined by both its properties and its interactions [36, 37, 38]. For instance, in a gene regulatory network, the activity of a gene is influenced by its regulatory elements and the interacting proteins. Graph Convolutional Networks (GCNs) are a subset of GNNs that aggregate information through convolutions [39, 33]. Convolutions involve a linear aggregation of features from

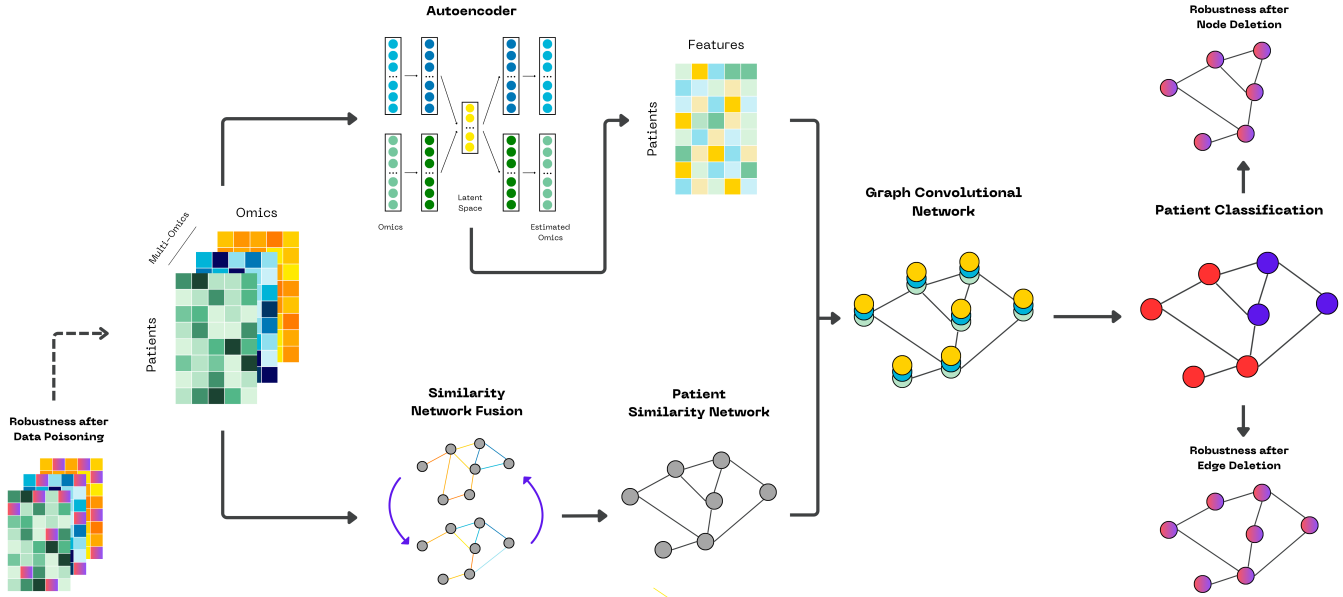


Fig. 1: Schematic of the model’s workflow. The multi-omics datasets are the initial input. The Autoencoder is used to capture meaningful variation across omics while creating a matrix of lower dimensions that is used as a features matrix by the GCN. The SNF is used to construct a PSN that captures similarity in omics variation between patients. This pattern of connections forms the input adjacency matrix to the GCN. The features matrix and adjacency matrix are fed into the GCN, which is trained to perform node (i.e., patient) classification. The Autoencoder to Patient Classification analyses are part of the MoGCN framework. Finally, three adversarial attacks involving node deletion, edge deletion, and data poisoning are used to test its Robustness.

neighbors, followed by a weighted transformation and a nonlinear activation function. GCNs offer the advantages of simplicity and efficiency, and have been successfully implemented in multi-omics integration [40, 41, 42, 43].

Nevertheless, deep-learning models including GCNs also have disadvantages. A key challenge is the limited availability of large datasets, particularly for randomized controlled trials (RCTs). Small samples can lead to overfitting, a situation where models perform well on training data but fail to generalize to new data [34]. It is not known how the performance of GNNs in multi-omics integration may be affected by a small sample size, given that previous research uses data from large databases [36, 40, 41, 44]. Additionally, deep learning models are highly sensitive to input variations, meaning that small changes can lead to different predictions [45]. This can be an issue for robustness, i.e., a model’s ability to perform at a similar capacity when provided with noisy data. One of the techniques used to estimate the robustness of deep-learning models is Adversarial Robustness [46]. Adversarial robustness involves intentionally introducing small perturbations to the input data, known as “attacks”, to test the model’s resilience. A robust model can learn meaningful representations from the data, whereas a model that is not robust cannot distinguish meaningful patterns from noise [47]. Knowing when or why a model fails an adversarial attack can inform the development of an alternative model.

I consider a deep learning model that includes a GCN on multi-omics data obtained from an RCT and subsequently perform tests of Adversarial Robustness with two specific goals: (1) evaluate the performance of GCNs with small clinical samples and compare that with the performance of GCNs on large database samples

found in the literature; (2) investigate the robustness of GCNs to assess changes in performance when presented with perturbed multi-omics data. To achieve these goals, I utilize multi-omics data from participants in the TOFA-predict study, an RCT study conducted to assess the efficacy of tofacitinib, a medication for the treatment of Psoriatic Arthritis (PsA) [48]. PsA is a heterogeneous inflammatory disease whose pathobiology is poorly understood; thus, omics data are tools used to explore its underlying biological processes [48]. The study includes two patient groups: newly diagnosed patients, dubbed naïve, and previously diagnosed patients who received pharmacological treatment but did not respond to it, dubbed resistant. In the 4-arm RCT half of each group is randomly assigned to Tofacitinib or alternative treatment and there are multiple follow-up moments. However, I focus exclusively on the multi-omics data that was collected at baseline. The primary objective of this research is to test whether the resistant patients exhibit distinct data patterns. If the resistant patients can be successfully identified using multi-omics data, medical professionals could use insights from the model’s interpretation to identify potential biomarkers for response to pharmacotherapy and to understand the molecular mechanisms of RA and its progression.

## Methods

I consider the MoGCN framework for data analysis and prediction [44]. MoGCN is a deep learning model that utilizes multi-omics data for node classification, where a node may represent a patient or a sample. It consists of multiple components: an Autoencoder, a Similarity Network Fusion, and a GCN. Following the prediction task, the robustness of the model against adversarial attacks is

investigated. The analyses are performed in Python with Pandas, Pytorch, and Snfpy.

### Data

The omics data were measured from plasma samples and included one proteomics profile and four metabolomics profiles. The proteomics data were measured with Olink Proteomics technology, yielding 368 proteins. The metabolomics profiles were measured with UPLC-MS on four platforms: Amino acids, Signaling Lipids, Lipids, and Global. In the Amines profile, 57 compounds were identified using Study Quality Control (SQC) correction. For the Signaling Lipids profiling at high pH, 70 compounds were reported using SQC correction. For the Signaling Lipids profiling at low pH, 92 compounds were identified using SQC correction. Currently, the data for the Lipids and Global Metabolomics is still being processed.

### Autoencoder

An autoencoder is a NN architecture used for unsupervised learning [49, 50]. It’s designed to compress the input data into a lower-dimensional representation and then reconstruct it back to the original form. It consists of two components: an encoder  $f$  and a decoder  $g$ , which function in tandem. The encoder transforms data from its original domain to a compressed representation in a latent space  $Z$  with lower dimensions  $L$ . Subsequently, the decoder reconstructs the data back into the original space from this latent space, thus  $z = f(x)$  and  $\tilde{x} = g(z)$ . This architecture is designed to identify and retain critical features of the data by minimizing a reconstruction loss, which is a measure of how well the output of the autoencoder resembles the original input. For multi-omics, an autoencoder can reduce the dimensionality of the data, extracting the most relevant features while minimizing information loss and reducing noise.

In MoGCN [44], the autoencoder is used to “condense” the multi-omics data from a three-dimensional structure (i.e., gene  $\times$  patients  $\times$  omics) into a two-dimensional matrix, which is later needed in the GCN. Thus, even though both the encoder and decoder are optimized, the focus is on the latent representations generated by the encoder, which reduce the multi-omics dimensions. The input data comprise five types of omics data, which are represented by the matrices  $X_1, \dots, X_5$ . Because there is more than one input matrix, MoGCN utilizes a multi-modal autoencoder architecture. This structure consists of several encoders and decoders, each corresponding to an omic type, but all converging to a shared latent layer. The loss function in this multi-modal setup is an aggregate of weighted losses for each data type:

$$E = \operatorname{argmin}_{f,g} (\alpha \operatorname{Loss}_1(x_1, g_1(f_1(x_1))) + \dots + \epsilon \operatorname{Loss}_5(x_5, g_5(f_5(x_5))), \quad (1)$$

where  $\alpha, \dots, \epsilon$  are weights that reflect the relative importance or prior knowledge of each omics type, and they sum up to 1. This weighted approach can accommodate the nuances of each omics type.

### Similarity Network Fusion and Patient Similarity Network

The Similarity Network Fusion (SNF) algorithm is designed to integrate different types of data by creating a network or

graph where each node represents a sample and edges represent similarities between these samples [51]. The SNF can be used to compute and combine Patient Similarity Networks (PSNs). First, the SNF calculates within-patient similarity matrices across omics. For  $n$  patients the results are  $n$  similarity matrices across omics. Then, these matrices are fused to construct a network that maps the similarities between patients. This process involves amplifying the strong connections and diminishing the weaker ones. The outcome of the fusion is one PSN that captures patterns across omics and patients.

In MoGCN [44], the fused PSN is what defines the structure of the graph for the GCN. Each patient is represented as a node and edges placed between patients represent similar omics patterns, effectively creating a network that mirrors the biological similarities between patients. With one patient  $i$  and two omics  $m, n$ , the equation of the scaled exponential patient similarity matrix is:

$$W(i, i) = \exp \left( -\frac{\rho^2(x_m, x_n)}{\mu \epsilon_{m,n}} \right), \quad (2)$$

where the  $n \times n$  patient-patient similarity matrix is  $W(i, j)$ ,  $\rho(x_m, x_n)$  is the Euclidean distance between the patients’ omics  $x_m$  and  $x_n$ ,  $\mu$  is a hyperparameter that can be empirically set, and  $\epsilon_{m,n}$  is a hyperparameter used to solve the problem of scaling. Different omics datasets have largely different scales, so when calculating the Euclidean distance it’s important to include a correction to prevent the larger scales from disproportionately influencing the distance calculation.

Then, the similarity matrix of two patients  $P$  and K-nearest similarity matrix  $S$  can be defined as

$$P(i, j) = \begin{cases} \frac{W(i, j)}{2 \sum_{k \neq i} W(i, k)} & \text{if } j \neq i \\ \frac{1}{2} & \text{otherwise,} \end{cases} \quad (3)$$

$$S(i, j) = \begin{cases} \frac{W(i, j)}{\sum_{k \in N_i} W(i, k)} & \text{if } j \in N_i, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

When dealing with similarity matrices for multi-omics data, the matrices can be quite dense, with non-zero values indicating some level of similarity between all pairs of patients. However, these similarities are not all equally relevant, as a patient is more likely to be closely related to a small subset of other patients rather than all patients. The K-nearest similarity matrix is a version of the original matrix where for each patient only the similarities with their K-nearest neighbors are retained, and other connections are set to zero. This makes the matrix sparser and more focused. The similarity matrices can be iteratively updated to account for all patients. For the formulas describing this iterative process and the generalization to  $m$  omics, readers can refer to the original MoGCN paper.

### Graph Convolutional Network

The core concept of GNN architectures revolves around the iterative updating of node representations by integrating the representations of adjacent nodes with their own. At each layer of the GNN, information from the neighbors of each node is merged with an *AGGREGATE* function and subsequently integrated with a *COMBINE* function [33]. With  $k$  layers,  $\nu = 1, \dots, n$  nodes, and

$N(\nu)$  denoting the neighbors of a node, the graph state  $H$  is defined as:

Initialization:  $H^0 = X$

For  $k = 1, 2, \dots, K$ ,

$$a_\nu^k = \mathbf{AGGREGATE}^k \{H_u^{k-1} : u \in N(\nu)\}$$

$$H_\nu^k = \mathbf{COMBINE}^k \{H_\nu^{k-1}, a_\nu^k\}.$$

After each layer, the combined representations become the new node representations. After the final layer, node representations are used for downstream tasks, for example node classification. The predicted label of a node  $\hat{y}_i$  can be computed with a Softmax function. Training a GNN model involves minimizing a loss function, like cross-entropy. The network is optimized using backpropagation to minimize the objective function, in this case the discrepancy between predicted and observed node labels. GCNs are a subset of GNNs where the node representations in each layer are updated according to a propagation rule designed to approximate convolutional filters, which helps in linearizing the graph structure for computation [39].

In MoGCN [44], the nodes of the GCN each represent one patient, while the edges between them represent similarity between patients as calculated with the PSN. The node features are the output of the Autoencoder, meaning that each patient is described using the latent representation of their multi-omics data. At each GCN layer, each patient's features are updated as such: (1) the latent representation of their omics data is aggregated with those of similar patients, and (2) the aggregated representations go through a convolution, creating a single new feature representation. At the final layer, each patient is assigned a label to classify them as naïve or resistant.

To assign patients labels, the GCN algorithm requires a graph structure and node features [39]. The node features are contained in the feature matrix  $X$ , whose dimensions are  $n$  (the number of nodes, i.e., patients), and  $f$  (the number of features per node). The graph structure, i.e., its connections, are defined by adjacency matrix  $A$ . The GCN is constructed with multiple convolutional layers, each defined by the equation

$$H^{k+1} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^k W^k \right), \quad (5)$$

where  $\tilde{A} = A + I$  is the adjacency matrix  $A$  plus self-connections  $I$ ;  $\tilde{D}$  is a diagonal matrix with  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ;  $\sigma(\cdot)$  is an activation function like ReLU;  $H^k$  is the graph state; and  $W^k$  is a linear transformation matrix to be trained [33].

### Adversarial Robustness

Investigating GCN robustness involves creating worst-case perturbations, i.e., minor data changes that lead to substantial output differences [52]. Adversarial robustness can be framed as an optimization problem in which the goal is to find data perturbations that maximize certain attack objectives, such as promoting an incorrect classification [46]. Adversarial attacks can vary depending on their scenario, type, and perturbation space [46].

There are two scenarios: poisoning and evasion. With poisoning, perturbations occur before the model's training, impacting both

the learning process and the resulting model. Evasion involves altering data during the testing phase, with the model already trained and fixed. Attack types can be local or global. Local attacks focus on how predictions for a specific node change under perturbations, while global attacks assess the impact on overall performance. The perturbation space determines what can be changed and to what extent. Changes can involve adding or removing edges, adding or removing nodes, changing node attributes, and changing classification labels. Adversarial attacks are typically designed to be indistinguishable from the original input to the human eye. For a graph, defining "nearly indistinguishable" is challenging because humans can readily notice changes. Previous research focuses on budget constraints which limit the number of changes in an attack [53, 45]. These constraints can be local, like limiting the number of edge deletions per node, or global, like limiting the total number of edge deletions across the graph.

To assess robustness, I implement three poison attacks, each designed to exploit different vulnerabilities of the analysis process. The first attack involves removing 5% of connections in the GCN graph [54]. This simulates errors that might occur during the construction of the PSN, which could impact the network's structure and the subsequent learning process of the GCN. The second attack involves removing 2% of nodes from the network [55]. This simulates scenarios of missing data or patient dropout, a common challenge in real-world clinical trial datasets. These attacks were selected based on previous research and their effect is measured based on the model's global performance. These attacks poison the input directly to the GCN allowing for an assessment of the GCN's robustness. However, introducing poisoned data at the start, before the processing by the Autoencoder and PSN, enables an evaluation of the overall MoGCN framework. Therefore, the third attack involves corrupting the omics datasets, simulating errors during measurements, or raw data preprocessing. Such corruption may lead to biased data being fed into the GCN, thereby also testing its ability to handle inaccurately encoded features.

### Table

Table 1 contains simulated data and is an example of what can be expected from the final thesis.

**Table 1.** Original and MoGCN-predicted labels.

	Predicted: Naïve	Predicted: Resistant
Observed: Naïve	31	9
Observed: Resistant	4	36

Example of the Confusion Matrix of predicted labels by observed labels. The numbers are estimates obtained from simulated data.

## Competing interests

No competing interest is declared.

## Author contributions statement

G.O. conceived and conducted the experiment, G.O. and S.B. analysed the results. G.O. wrote the manuscript, G.O., S.B., and H.W.U reviewed the manuscript.

## Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.

## References

1. Biswapriya B Misra, Carl Langefeld, Michael Olivier, and Laura A Cox. Integrated omics: tools, advances and future approaches. *Journal of molecular endocrinology*, 62(1):R21–R45, 2019.
2. Bruno Aranda, P Achuthan, Yasmin Alam-Faruque, I Armean, Alan Bridge, C Derow, Marc Feuermann, AT Ghanbarian, Samuel Kerrien, Jyoti Khadake, et al. The intact molecular interaction database in 2010. *Nucleic acids research*, 38(suppl.1):D525–D531, 2010.
3. Heather J Cordell. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, 10(6):392–404, 2009.
4. Mario Vailati-Riboni, Valentino Palombo, and Juan J Loor. What are omics sciences? *Periparturient diseases of dairy cows: a systems biology approach*, pages 1–7, 2017.
5. Efi Athieniti and George M Spyrou. A guide to multi-omics data collection and integration for translational medicine. *Computational and Structural Biotechnology Journal*, 2022.
6. Said el Bouhaddani, Matthias Höllerhage, Hae-Won Uh, Claudia Moebius, Marc Bickle, Günter Höglinger, and Jeanine Houwing-Duistermaat. Statistical integration of multi-omics and drug screening data from cell lines. *bioRxiv*, 2022-08, 2023.
7. Indhupriya Subramanian, Srikant Verma, Shiva Kumar, Abhay Jere, and Krishanpal Anamika. Multi-omics data integration, interpretation, and its application. *Bioinformatics and biology insights*, 14:1177932219899051, 2020.
8. Hassan Dihazi, Abdul R Asif, Tim Beißbarth, Rainer Bohrer, Kirstin Feussner, Ivo Feussner, Olaf Jahn, Christof Lenz, Andrzej Majcherczyk, Bernhard Schmidt, et al. Integrative omics-from data to biology. *Expert review of proteomics*, 15(6):463–466, 2018.
9. Yehudit Hasin, Marcus Seldin, and Aldons Lusi. Multi-omics approaches to disease. *Genome Biology*, 18:83, 2017.
10. Ran Duan, Lin Gao, Yong Gao, Yuxuan Hu, Han Xu, Mingfeng Huang, Kuo Song, Hongda Wang, Yongqiang Dong, Chaoqun Jiang, et al. Evaluation and comparison of multi-omics data integration methods for cancer subtyping. *PLoS computational biology*, 17(8):e1009224, 2021.
11. Sonia Guil and Manel Esteller. Rna–rna interactions in gene regulation: the coding and noncoding players. *Trends in biochemical sciences*, 40(5):248–256, 2015.
12. Michael M Hann, Andrew R Leach, and Gavin Harper. Molecular complexity and its impact on the probability of finding leads for drug discovery. *Journal of chemical information and computer sciences*, 41(3):856–864, 2001.
13. Konrad J Karczewski and Michael P Snyder. Integrative omics for health and disease. *Nature Reviews Genetics*, 19(5):299–310, 2018.
14. Oscar Méndez-Lucio and José L Medina-Franco. The many roles of molecular complexity in drug discovery. *Drug Discovery Today*, 22(1):120–126, 2017.
15. Mengyun Wu, Huangdi Yi, and Shuangge Ma. Vertical integration methods for gene expression data analysis. *Briefings in Bioinformatics*, 22, 5 2021.
16. Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2):bbab569, 2022.
17. Amrit Singh, Casey P Shannon, Benoît Gautier, Florian Rohart, Michaël Vacher, Scott J Tebbutt, and Kim-Anh Lê Cao. Diablo: an integrative approach for identifying key molecular drivers from multi-omics assays. *Bioinformatics*, 35(17):3055–3062, 2019.
18. Milan Picard, Marie-Pier Scott-Boyer, Antoine Bodein, Olivier Périn, and Arnaud Droit. Integration strategies of multi-omics data for machine learning analysis. *Computational and Structural Biotechnology Journal*, 19:3735–3746, 2021.
19. Parminder S Reel, Smarti Reel, Ewan Pearson, Emanuele Trucco, and Emily Jefferson. Using machine learning approaches for multi-omics data analysis: A review. *Biotechnology Advances*, 49:107739, 2021.
20. Priya Tolani, Srishti Gupta, Kirti Yadav, Suruchi Aggarwal, and Amit Kumar Yadav. Big data, integrative omics and network biology. *Advances in Protein Chemistry and Structural Biology*, 127:127–160, 2021.
21. Yuqi Wen, Linyi Zheng, Dongjin Leng, Chong Dai, Jing Lu, Zhongnan Zhang, Song He, and Xiaochen Bo. Deep learning-based multiomics data integration methods for biomedical application. *Advanced Intelligent Systems*, page 2200247, 2023.
22. Eva Brombacher, Maren Hackenberg, Clemens Kreutz, Harald Binder, and Martin Treppner. The performance of deep generative models for learning joint embeddings of single-cell multi-omics data. *Frontiers in Molecular Biosciences*, 9, 10 2022.
23. Dongjin Leng, Linyi Zheng, Yuqi Wen, Yunhao Zhang, Lianlian Wu, Jing Wang, Meihong Wang, Zhongnan Zhang, Song He, and Xiaochen Bo. A benchmark study of deep learning-based multi-omics data fusion methods for cancer. *Genome biology*, 23(1):1–32, 2022.
24. Li Tong, Jonathan Mitchel, Kevin Chatlin, and May D Wang. Deep learning based feature-level integration of multi-omics data for breast cancer patients survival analysis. *BMC medical informatics and decision making*, 20:1–12, 2020.
25. Zhi Huang, Xiaohui Zhan, Shunian Xiang, Travis S Johnson, Bryan Helm, Christina Y Yu, Jie Zhang, Paul Salama, Maher Rizkalla, Zhi Han, et al. Salmon: survival analysis learning with multi-omics neural networks on breast cancer. *Frontiers in genetics*, 10:166, 2019.
26. Giovanna Nicora, Francesca Vitali, Arianna Dagliati, Nophar Geifman, and Riccardo Bellazzi. Integrated multi-omics analyses in oncology: a review of machine learning methods and tools. *Frontiers in oncology*, 10:1030, 2020.

27. Peng Cui, Lingfei Wu, Jian Pei, Liang Zhao, and Xiao Wang. Graph representation learning. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 17–26, 2022.
28. Marianna Milano and Mario Cannataro. Network models in bioinformatics: modeling and analysis for complex diseases. *Briefings in Bioinformatics*, 24:1–2, 2023.
29. Jiening Zhu, Jung Hun Oh, Anish K. Simhal, Rena Elkin, Larry Norton, Joseph O. Deasy, and Allen Tannenbaum. Geometric graph neural networks on multi-omics data to predict cancer survival outcomes. *Computers in Biology and Medicine*, 163, 9 2023.
30. Chunyu Wang, Junling Guo, Ning Zhao, Yang Liu, Xiaoyan Liu, Guojun Liu, and Maozu Guo. A cancer survival prediction method based on graph convolutional network. *IEEE Transactions on NanoBioscience*, 19:117–126, 2020.
31. Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. Moronet: multi-omics integration via graph convolutional networks for biomedical data classification. *bioRxiv*, pages 2020–07, 2020.
32. Pan Li and Jure Leskovec. *The Expressive Power of Graph Neural Networks*, pages 63–98. Springer Nature Singapore, Singapore, 2022.
33. Jian Tang and Renjie Liao. Graph neural networks for node classification. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 41–61, 2022.
34. Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Le Song. *Graph Neural Networks*, pages 27–37. Springer Nature Singapore, Singapore, 2022.
35. Yanhui Zhang, Xiaohui Lin, Zhenbo Gao, Tianxiang Wang, Kunjie Dong, and Jianjun Zhang. An omics data analysis method based on feature linear relationship and graph convolutional network. *Journal of Biomedical Informatics*, 145:104479, 9 2023.
36. Sara Althubaiti, Maxat Kulmanov, Yang Liu, Georgios V Gkoutos, Paul Schofield, and Robert Hoehndorf. Deepmocca: A pan-cancer prognostic model identifies personalized prognostic markers through graph attention and multi-omics data integration. *BioRxiv*, 2021-03, 2021.
37. Qianqian Song, Jing Su, and Wei Zhang. scgcn is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nature communications*, 12:3826, 2021.
38. Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. Mognet integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12:3445, 2021.
39. Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
40. Qiao Liu, Zhiqiang Hu, Rui Jiang, and Mu Zhou. Deepcdr: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics*, 36:911–918, 2020.
41. Bingjun Li, Tianyu Wang, and Sheida Navabi. Cancer molecular subtype classification by graph convolutional networks on multi-omics data. Association for Computing Machinery, 2021.
42. Hongli Gao, Bin Zhang, Long Liu, Shan Li, Xin Gao, and Bin Yu. Problem solving protocol a universal framework for single-cell multi-omics data integration with graph convolutional networks. *Briefings in Bioinformatics*, 24:1–11, 2023.
43. Roman Schulte-Sasse, Stefan Budach, Denes Hnisz, and Annalisa Marsico. Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nature Machine Intelligence*, 3:513–526, 2021.
44. Xiao Li, Jie Ma, Ling Leng, Mingfei Han, Mansheng Li, Fuchu He, and Yunping Zhu. Mogn: A multi-omics integration method based on graph convolutional network for cancer subtype analysis. *Frontiers in Genetics*, 13, 2 2022.
45. Daniel Zügner, Oliver Borchert, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5):1–31, 2020.
46. Stephan Günnemann. *Graph Neural Networks: Adversarial Robustness*, pages 149–176. Springer Nature Singapore, Singapore, 2022.
47. Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.
48. Nienke J Kleinrensink, Frank T Perton, Juliëtte N Pouw, Nanette LA Vincken, Sarita AY Hartgring, Mylène P Jansen, Saeed Arbabi, Wouter Foppen, Pim A de Jong, Janneke Tekstra, et al. Tofa-predict study protocol: a stratification trial to determine key immunological factors predicting tofacitinib efficacy and drug-free remission in psoriatic arthritis (psa). *BMJ open*, 12(10):e064338, 2022.
49. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
50. Vanessa Böhm and Uroš Seljak. Probabilistic autoencoder. *arXiv preprint arXiv:2006.05479*, 2020.
51. Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333–337, 2014.
52. Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2847–2856, 2018.
53. Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. Towards more practical adversarial attacks on graph neural networks. *Advances in neural information processing systems*, 33:4756–4766, 2020.
54. Daniel Zügner and Stephan Günnemann. Certifiable robustness of graph convolutional networks under structure perturbations. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1656–1665, 2020.
55. Ben Finkelshtein, Chaim Baskin, Evgenii Zheltonozhskii, and Uri Alon. Single-node attacks for fooling graph neural networks. *Neurocomputing*, 513:1–12, 2022.