ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Designing meaningful measures to evaluate generative graph neural networks on protein datasets.

Master Thesis

Philip Jean Hartout

July 10, 2022

Advisors: Prof. Dr. Karsten M. Borgwardt, Tim Kucera
Department of Biosystems Science and Engineering, ETH Zürich

Abstract

This example thesis briefly shows the main features of our thesis style, and how to use it for your purposes.

Contents

Chapter 1

Introduction

Chapter 2

Background & Related Work

This chapter introduces the core concepts built upon in this thesis and surveys recent literature tackling the evaluation of generative graph neural networks (GNNs) and the relevance of this problem in structural biology. Section [**REVISE**] defines core mathematical and biological concepts that will be built upon in the thesis. Section [**REVISE**] will discuss recent advances in the design of measures used to evaluate generative GNNs and in structural biology.

The set of methods investigated in this thesis lies at the interface of structural biology and machine learning. We start by defining some relevant biological properties of proteins, followed by a survey various graph theoretical abstractions derived from the protein structure. We then move on to define generative models and the various classes of measures used to evaluate them.

2.1 Proteins

Proteins are large biomolecules that are formed from a sequence of amino acids, performing their functions as determined by their three-dimensional structure, and amino acid sequence. Proteins support a vast array of functions in living organisms, such as catalysing metabolic reactions, DNA replication, providing structural support to cells, transporting molecules and sensing stimuli.

Each protein is made up of one or more chains of amino acids, each of which contain a backbone and different side chains. The atoms in the backbone include a α-carbon, another carbon and a nitrogen atom. An overview of the peptide backbone is shown in Figure 2.1. Interestingly, a plane is forned by two alpha carbons, the carboxyl group, and the hydrogen atom attached to the nitrogen atom (see Figure 2.1), making the peptide bond between

the nitrogen and carbon atom resistant to twisting. That means that the rotations enabling the 3D folding of a protein is governed by the angle of the bonds linking the nitrogen atom to the α-carbon and the other carbon atom to the α-carbon, named φ and ψ. These angles' values are frequently used to validate proteins, characterise the secondary structure of proteins (i.e. structural features observed in certain segments of proteins), etc.

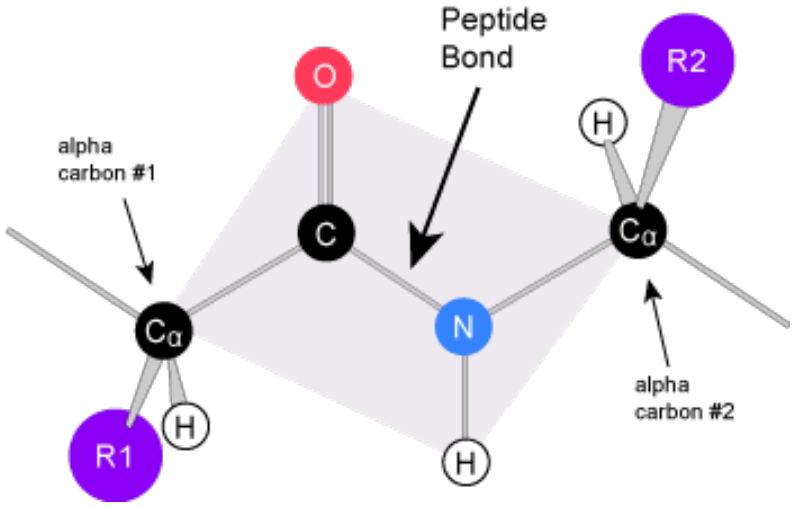

Figure 2.1: Schematic of the backbone of a protein. Two α-carbons are shown as well as a β-carbon in the middle. R1 and R2 represent the side chains of the amino acid.

To visualize such angles, a Ramachandran plot can be constructed for any protein. Such a plot can reveal secondary structural features such as β-sheets, α-helices, etc. An example of such a plot together with a 3D model of a protein can be found in Figure 2.2.

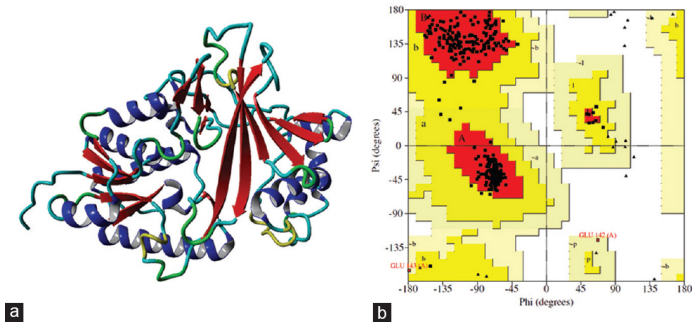

Figure 2.2: 3D structure of uridine diphosphogalactofuranose-galactopyranose mutase with a corresponding Ramachandran plot. The α-helices can be found on the middle left part of the Ramachandran plot, the β sheets on the upper right quadrant, and the left handed α-helices can be found in the middle upper right part of the plot. This figure is adapted from Nayak et al. [2018].

2.2 Graphs

Proteins are often abstracted using graphs. A graph G is a pair of vertices V and edges E such that $G = (V, E)$, $|V| = n$ and $|E| = m$. Two vertices i and j are adjacent if there is an edge between them, i.e. $e_{ij} \in E$. The relationship between between edges can be represented as an $n \times n$ adjacency matrix A, where:

$$A_{ij} = \begin{cases} 1, & \text{if } e_{ij} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The neighborhood of a node v is the set of nodes with an edge directly to v, i.e. $N(v) = \{u \in V | e_{uv} \in E\}$. A graph is undirected if the edges do not contain directional information, i.e. $A_{ij} = A_{ji}$. A directed graph would result in directionality being encoded in edges, where A_{ij} would not contain any information about A_{ji}. Nodes and edges in each graph can contain one or more labels. In this thesis, we will mostly deal with labeled undirected graphs, where each node will be labeled according to the amino acid type each node belongs to.

There are multiple ways of constructing graphs from proteins. First, one can extract a *contact map* of a protein by computing the (euclidean) distance between any two points belonging to each amino acid. The α-carbon is often used for this purpose. This is a fully connected graph with continuously labeled edges representing the distance between each node. From there, it is possible to either extract a k-nearest neighbour graph, where $k \in \mathbb{N} > 0$ defines the amount of nodes directly connected to any given nodes; or an ε-graph, where each node within a given distance $\varepsilon \in \mathbb{R}^+ \setminus \{0\}$ of another node is connected. Both are graphs where each node is labeled with the residue name to which the α-carbon belongs and the edges are unlabeled.

2.3 Topological Data Analysis

Although graphs are powerful representation of proteins, the latter can also be represented as *point clouds*. One powerful field of study of topological properties of point clouds (among other structured data) is *topological data analysis*.

Topology has witnessed relentless theoretical progress since Henri Poincaré first addressed topological ideas as a distinct branch of mathematics in his 1895 publication of *Analysis Situs* [Poincaré, 1895]. Only recently, – with the advent of modern computing – has the field of computational topology and topological data analysis (TDA) gained momentum to investigate (high-dimensional) data in physics, biology, and beyond [Dey et al., 1999, Ghrist, 2008, Amézquita et al., 2020]. For material providing an extensive and formal introduction to topology and persistent homology, please refer

to [Freedman and Chen, 2009, Edelsbrunner and Harer, 2010], and [Ghrist, 2008].

A powerful computational technique to analyse topological properties of point clouds is *persistent homology*, which first requires us to define simplicial homology. Simplicial homology refers to a way of assigning connectivity information to topological objects, such as point clouds, which are represented by simplicial complexes. A simplicial complex K is a set of simplices which correspond to vertices in dimension 0, edges in dimension 1 and triangles in dimension 2. The subsets of a simplex $\sigma \in K$ are referred to as its faces, and every face $\tau \in K$. Moreover, any non-empty intersection of two simplices also needs to be part of the simplicial complex, i.e. $\sigma \cap \sigma' \neq \emptyset$ for $\sigma, \sigma' \in K$ implies $\sigma \cap \sigma' \in K$, meaning that K is closed under calculating the faces of a simplex.

Persistent homology extends simplicial homology by employing filtrations to imbue K with scale information. This process captures rich, mutli-scale topological information related to K in a principled way. The filtration process is generally defined by a function $f : K \to \mathbb{R}$ satisfying some finite number of values m and $f^0 \leq f^1 \leq \cdots \leq f^{m-1} \leq f^m$. This allows us to sort K using f, for instance by extending f linearly to higher-dimensional simplices via $f(\sigma) := \max_{v \in \sigma} f(v)$, leading to a nested sequence of simplicial complexes like so:

$$\emptyset = K^{(0)} \subseteq K^{(1)} \subseteq \cdots \subseteq K^{(m-1)} \subseteq K^{(m)}, \tag{2.2}$$

where $K^{(i)} := \{\sigma \in K \mid f(\sigma) \leq f^{(i)}\}$. This relationship enables tracking the appearance (i.e. a connected component arising) and the dissapearance (i.e. two connected components into one) of topological features across scales as one transitions from $K^{(i)}$ to $K^{(i+1)}$. The birth (i.e. appearance) and death (i.e. dissapearance) of topological features for different values of f are usually summarized in a *persistence diagram*, which is a multiset of tuples, each of which contains the values at which each features is born or dies.

A common construction for obtaining such features is the Vietoris-Rips complex [Vietoris, 1927]. It requires a distance threshold ε and a metric $(\cdot, \cdot)$ (usually, the Euclidean distance, as we will use in this thesis). The Vietoris-Rips complex at scale ε of an input protein point cloud is defined as $\mathcal{V}_\varepsilon(X) := \{\sigma \subseteq X | (x(i), x(j)) \leq \varepsilon\}$, $\forall x(i), x(j) \in \sigma$, i.e. $\mathcal{V}_\varepsilon$ contains all subsets of the input space whose pairwise distances are less than or equal to ε. $\mathcal{V}_\varepsilon$ is conceptually very similar to the ε-graphs discussed in section 2.2, except that ε here ranges over the entire space of possible distance values, and $\mathcal{V}_\varepsilon$ also tracks topological features over all three dimensions, instead of

only connected nodes[1].

Note that the multiplicity of the persistence diagram corresponds to the number of homology dimensions under study. In this thesis, given proteins are represented as three-dimensional point clouds, we choose to track topological features across three homology dimension: 0,1 and 2. Effectively, this tracks connected components in dimension 0, circular holes in dimension 1 and two dimensional voids or cavities in dimension 2 as the filtration function is applied. For a more thorough introduction to homology and homology groups, please refer to Edelsbrunner and Harer [2010].

2.4 Generative models

This thesis deals with measures to assess generative model performance, so we define generative models here. While discriminative machine learning techniques aim to learn some dependent variable $\mathcal{Y}$ from a set of (independent) features $\mathcal{X}$, generative machine learning models generate synthetic samples $\mathcal{X}'$ following the distribution of $\mathcal{X}$. Computing such probabilistic distributions through maximum likelihood estimation and related methods is intractable in many cases; as such, new learning paradigms were established to enable the modeling of complex, real-world distributions through gradient-based methods.

One such seminal method was that of generative adversarial learning, pioneered by Goodfellow et al. [2014], where a (deep) generator is pitted against a (deep) discriminator. The former's goal is to generate samples identical to the training distribution, while the latter is to classify whether or not the sample originated from the generator or the training distribution. Simultaneously developed methods by Kingma and Welling [2013] generalized this idea further and introduced the variational auto-encoder, where instead of a discriminator, the second network leverages the representation of the generator to perform approximate inference. In both cases, the two networks (i.e. the generator and the discriminator/inference network) are jointly trained using backpropagation to minimize some appropriate loss function.

[ADD LOSS FUNCTIONS]

A recent review of the existing landscape of generative modelling methods has been provided by Bond-Taylor et al. [2021].

These techniques have been particularly successful in the image domain, where modern GANs have been able to tackle multiple practical challenges such as mode collapse and convergence failure to produce realistic images,

[1] Technically, since our input consists of points (a.k.a. 0-simplicial complexes) exclusively (and no 1- or 2- simplicial complexes), we are actually primarily concerned with a subcomplex of $\mathcal{V}_\varepsilon(X)$ called the Čech complex.

Figure 2.3: Sample images generated by StyleGAN-XL, the state-of-the-art GAN by Sauer et al. [2022] at the time of writing.

such as the sample seen in Figure 2.3. More pertinent to this thesis is the application of generative models to graphs. The application domain has been reviewed by Zhou et al. [2020]. In short, graph generative networks are capable of operating on the highly versatile and extensible graph domain. It has been shown that they can produce small molecules, generate social networks, knowledge graphs, among many other real-world tasks. Generative networks can be grouped into two categories: those that generate nodes in each graph sequentially, such as GraphRNN by You et al. [2018], and those that generate graphs from some latent distribution directly, such as MolGAN by De Cao and Kipf [2018].

Operating in the graph domain incurs some unique challenges. From a modelling standpoint, dealing with graphs means dealing with a much larger and variable output space. In the general case, at least n^2 values need to be specified. Additionally, the number of edges and nodes vary from sample to sample, which also needs to be accounted for in the model structure. Additionally, building a generative model generating graphs of up to n nodes, $n!$ possible adjacency matrices can be generated. Such a high representation complexity is challenging to model, expensive to compute, and difficult for objective functions to optimize. The last modelling-related issue when dealing with graphs is that the presence of one edge is not independent from another, i.e. real-world graphs often exhibit patterns of local connectedness which need to be accounted for in the model.

2.5 The Evaluation Problem

But perhaps the most significant problem plaguing all generative models is the evaluation problem. Concretely, this problem can be framed using the following question: how does the practitioner go about evaluating the quality of the set of samples generated? While sidestepping the problem is possible in the image domain by manually inspecting generated samples, a practice that might reveal interesting modelling pathologies (see Figure

Figure 2.4: Class-conditional samples generated by StyleGAN3 (left) and StyleGAN-XL (right) trained on the same dataset at the same resolution. This figure is adapted from Sauer et al. [2022].

2.4), this cannot be done at scale, nor can it be done for generative models operating in the graph domain, where human perception cannot easily evaluate the quality of a set of generated graphs. The community has therefore devised a set of measures to attempt to rank models more adequately.

Before going through existing metrics, it is useful to state broad goals, or *desiderata* of metrics concerning generative modelling. As highlighted by O'Bray et al. [2021], (pseudo)-metrics must be endowed with the following properties:

1. **Expressivity**: Given two sets of samples $\mathcal{X}_1$ and $\mathcal{X}_2$, a suitable measure d should have $d(\mathcal{X}_1, \mathcal{X}_2)$ increasing monotonically as $\mathcal{X}_1$ and $\mathcal{X}_2$ become more and more dissimilar.

2. **Robustness**: $d(\mathcal{X}_1, \mathcal{X}_2)$ should be robust to small perturbations in either sets.

3. **Efficiency**: $d(\mathcal{X}_1, \mathcal{X}_2)$ should be fast to calculate should scale well with size and number of graphs.

For images, an interesting metric (and the current standard for that domain) is the Fréchet Inception Score, as introduced by Heusel et al. [2017]. Overall, the goal of this metric is to calculate some distance between the activations of a neural network feature computed from both the real-world images the network was trained on and the synthesized images. Concretely, this is achieved by calculating the squared Wasserstein metric between the generated and real representations computed from a neural neural network (commonly, the Inception v3 architecture from Szegedy et al. [2015] is used) as two multidimensional Gaussian distributions with parameters $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu_{rw}, \Sigma_{rw})$, respectively. The general formulation of the p^{th} Wasserstein distance between two distributions u and v is given by

$$W_p(u,v) := \left(\inf_{\gamma \in \Gamma(u,v)} \int_{M \times M} d(x,y)^p \, \mathrm{d}\gamma(x,y) \right)^{1/p}, \tag{2.3}$$

where (M,d) is a metric space, $\Gamma(u,v)$ denotes the collection of all measures on $M \times M$ with marginals u and v on the first and second factors, respectively. Intuitively, $W_p(u,v)$ can be interpreted as a generalization of the Minkowski distance between probability distributions instead of fixed-length vectors, the latter being given by:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}. \tag{2.4}$$

In the case of the Fréchet Inception Score, the squared Wasserstein distance between the Inception v3-derived representations of the images can be reformulated as follows:

$$\mathrm{FID} = ||\mu - \mu_{rw}||_2^2 + \mathrm{tr}(\Sigma + \Sigma_{rw} - 2(\Sigma^{1/2}\Sigma_{rw}\Sigma^{1/2})^{1/2}). \tag{2.5}$$

For the graph domain, such a measure is unfeasible due to the varying size of the output graphs and the lack of common consensus on embedding methods, partly due to the diversity of graph types, although progress is being made on this front, as reviewed by Xu [2021]. Interesting strides have been made in some domains, such as in the drug discovery field, where the penultimate layer of the ChemNet neural network can be used as input to the FID as showed by Preuer et al. [2018].

However, an interesting approach recently explored by Thompson et al. [2022] leverages the observation, made in part by Xu et al. [2018a], Morris et al. [2019], and Kipf and Welling [2016], that certain GNNs have the ability to extract meaningful representations without any training. Through a set of two perturbation experiments, similar to the work done by Xu et al. [2018b] and O'Bray et al. [2021], Thompson et al. [2022] show that using a randomly initialized Graph Isormorphism Network (GINs), first introduced by Xu et al. [2018a], provides a strong, domain-agnostic metric to evaluate generative GNNs. GINs –like the majority of GNNs for discriminative machine learning purposes– consist in (i) L propagation layers that perform some form of message passing between the nodes, computing rich representations of each node's neighbourhoods and (ii) some readout layer, aiming to compute some embedding and subsequent output. For GINs, the message passing layers computing each (hidden) node embedding v at layer l (denoted $\mathbf{h}_v^{(l)}$) is assigned the following value:

$$\mathbf{h}_v^{(l)} := \mathrm{MLP}^{(l)} \left(\mathbf{h}_v^{(l-1)} + f^{(l)} \left(\left\{ \mathbf{h}_v^{(l-1)} : u \in N(v) \right\} \right) \right), \tag{2.6}$$

$\forall v \in V$ where V is as defined in section 2.2, $\forall\ l > 0, \mathbf{h}_v^{(l)} \in \mathbb{R}^d$, $\text{MLP}^{(l)}$ is a multilayer perceptron, and $f^{(l)}$ is some aggregating function, such as mean, max or sum. The second part, i.e. the graph readout layer with skip connections, aggregates features from all nodes at each layer $l \in [1, L]$, concatenating them into one $(L \times d)$ dimensional vector x_i as follows:

$$\mathbf{x}_i = \text{CONCAT}\left(g\left(\left\{\mathbf{h}_v^{(l)} |\ v \in V\right\}\right) |\ l \in [1, L]\right) \tag{2.7}$$

where g can be chosen from the same set of functions as $f^{(l)}$. **[]**

While these developments are encouraging, practitioners designing generative GNNs such as Liao et al. [2019], Niu et al. [2020], and You et al. [2018] have generally gravitated towards the maximum mean discrepancy (MMD) measure to evaluate the quality of the graph.

2.6 Maximum Mean Discrepancy & Kernel Methods

A significant part of this thesis is centered around investigating the MMD statistic, so we define and examine existing around MMD research here. Introduced by Borgwardt et al. [2006] and further exposited by Gretton et al. [2012], this measures leverages the expressive power and versatility of *kernel functions* to evaluate distances between two sample distributions. What's more, Gretton et al. [2012] describe how this measure can be treated as a test statistic, from which a p-value can be computed, to test if two distributions are statistically significantly different from one another. MMD is therefore an ideal platform to leverage when trying to assess generative models.

First, we define *kernels*, an essential component of MMD. They measure the similarity of two sets of any structured object. Let $\mathcal{X}$ be a non-empty set, and a *kernel function* $\mathrm{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying the following properties:

- $\mathrm{k}(x_i, x_j) = \mathrm{k}(x_j, x_i)\ \forall\ x_i, x_j \in \mathcal{X}$
- $\sum_{i,j} c_i, c_j \mathrm{k}(x_i, x_j) \geq 0\ \forall\ x_i, x_j \in \mathcal{X}, \forall\ c_i, c_j \in \mathbb{R}$.

Given n samples from $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ and m samples from $Y = \{y_1, \ldots, y_m\} \subseteq \mathcal{X}$, the biased estimate of MMD^2 is given by:

$$\text{MMD}^2(X, Y) := \frac{1}{n^2} \sum_{i,j=1}^{n} \mathrm{k}(x_i, x_j) + \frac{1}{m^2} \sum_{i,j=1}^{m} \mathrm{k}(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{k}(x_i, y_j) \tag{2.8}$$

In accordance with Lemma 6 of Gretton et al. [2012], the diagonal elements of the first two kernel matrices in Equation 2.8 can be set to 0 to obtain an unbiased estimate of MMD.

In the graph domain, it is now incumbent upon the practitioner to (i) choose an appropriate (optional) graph descriptor and (ii) kernel with (iii) appropriate kernel hyperparameters. This process, along with its pitfalls and current practices, are discussed in more detail by O'Bray et al. [2021], but we want to give an overview of possible, common, and sensible choices for descriptor, kernel, and hyperparameter.

A common practice in the literature is to first extract some fixed-length graph representation using a range of commonly used descriptors such as:

- The degree histogram **[DEFINE]**
- The clustering coefficient histogram **[DEFINE]**
- The Laplacian spectrum histogram **[DEFINE]**

Once such representations are computed, it is possible to compute a kernel between any two such vectorized representations using the following kernel functions:

- The linear kernel **[DEFINE]**
- The Gaussian kernel **[DEFINE]**

We will neglect certain kernels used in the literature either because they are not positive semi-definite, such as the total variation kernel (see O'Bray et al. [2021], Appendix A1 for a proof), or because they bring capture little more information compared to existing accepted alternatives, they are inefficient to compute and not recommended to evaluate generative GNNs, e.g. the Earth mover's distance kernel [O'Bray et al., 2021].

We will, however, leverage two new classes of kernels that are applicable to models evaluating protein generative model performance. The first are graph kernels. In particular we will examine the most efficient to compute and rich graph descriptor used for biological data: the Weisfeiler-Lehmann kernel. The second class of kernels leveraged here operate directly on the persistence diagrams obtained using the filtration procedures described in section 2.3.

2.7 Summary

Chapter 3

Methods

Dummy text.

3.1 Example Section

Dummy text.

3.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Chapter 4

Results

Dummy text.

4.1 Example Section

Dummy text.

4.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Chapter 5

Discussion

Dummy text.

5.1 Example Section

Dummy text.

5.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Chapter 6

Conclusion

Dummy text.

6.1 Example Section

Dummy text.

6.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Appendix A

Dummy Appendix

You can defer lengthy calculations that would otherwise only interrupt the flow of your thesis to an appendix.

Bibliography

Erik J Amézquita, Michelle Y Quigley, Tim Ophelders, Elizabeth Munch, and Daniel H Chitwood. The shape of things to come: Topological data analysis and biology, from molecules to organisms. *Developmental Dynamics*, 2020.

Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *arXiv preprint arXiv:2103.04922*, 2021.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Tamal K Dey, Herbert Edelsbrunner, and Sumanta Guha. Computational topology. *Contemporary mathematics*, 223:109–144, 1999.

Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

Daniel Freedman and Chao Chen. Algebraic topology for computer vision. *Computer Vision*, pages 239–268, 2009.

Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L Hamilton, David Duvenaud, Raquel Urtasun, and Richard S Zemel. Efficient graph generation with graph recurrent attention networks. *arXiv preprint arXiv:1910.00760*, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Tapaswini Nayak, Lingaraja Jena, Pranita Waghmare, Bhaskar C Harinath, et al. Identification of potential inhibitors for mycobacterial uridine diphosphogalactofuranose-galactopyranose mutase enzyme: A novel drug target through in silico approach. *International Journal of Mycobacteriology*, 7(1):61, 2018.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pages 4474–4484. PMLR, 2020.

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. *arXiv preprint arXiv:2106.01098*, 2021.

Henri Poincaré. *Analysis situs*. Gauthier-Villars, 1895.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *arXiv preprint arXiv:2202.00273*, 2022.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. 2015. *arXiv preprint arXiv:1512.00567*, 2015.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W Taylor. On evaluation metrics for graph generative models. *arXiv preprint arXiv:2201.09871*, 2022.

Leopold Vietoris. Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. *Mathematische Annalen*, 97(1):454–472, 1927.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018a.

Mengjia Xu. Understanding graph embedding methods and their applications. *SIAM Review*, 63(4):825–853, 2021.

Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018b.

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.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.