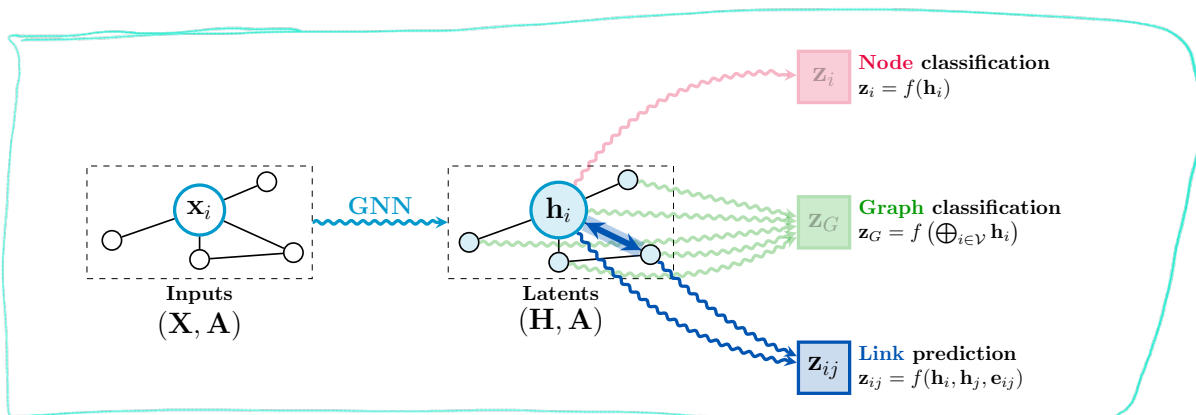# Everything is Connected: Graph Neural Networks

Petar Veličković

DeepMind / University of Cambridge

Abstract

In many ways, **graphs** are the main modality of data we receive from **nature**. This is due to the fact that most of the patterns we see, both in natural and artificial systems, are elegantly representable using the language of graph structures. Prominent examples include molecules (represented as graphs of atoms and bonds), social networks and transportation networks. This potential has already been seen by key scientific and industrial groups, with already-impacted application areas including traffic forecasting, drug discovery, social network analysis and recommender systems. Further, some of the most successful domains of application for machine learning in previous years—images, text and speech processing—can be seen as special cases of graph representation learning, and consequently there has been significant exchange of information between these areas. The main aim of this short survey is to enable the reader to assimilate the key concepts in the area, and position graph representation learning in a proper context with related fields.

1 Introduction: Why study data on graphs?

In this survey, I will present a vibrant and exciting area of deep learning research: graph representation learning. Or, put simply, building machine learning models over data that lives on *graphs* (interconnected structures of *nodes* connected by *edges*). These models are commonly known as *graph neural networks*, or **GNNs** for short.

There is very good reason to study data on graphs. From the molecule (a graph of *atoms* connected by chemical *bonds*) all the way to the connectomic structure of the brain (a graph of *neurons* connected by *synapses*), graphs are a universal language for describing living organisms, at all levels of organisation. Similarly, most relevant artificial constructs of interest to humans, from the transportation network (a graph of *intersections* connected by *roads*) to the social network (a graph of *users* connected by *friendship links*), are best reasoned about in terms of graphs.

This potential has been realised in recent years by both scientific and industrial groups, with GNNs now being used to discover novel potent antibiotics (Stokes et al., 2020), serve estimated travel times in Google Maps (Derrow-Pinion et al., 2021), power content recommendations in Pinterest (Ying et al., 2018) and product recommendations in Amazon (Hao et al., 2020), and design the latest generation of machine learning hardware: the TPUv5 (Mirhoseini et al., 2021). Further, GNN-based systems have helped mathematicians uncover the hidden structure of mathematical objects (Davies et al., 2021), leading to new top-tier conjectures in the area of representation theory (Blundell et al., 2021). It would not be an understatement to say that billions of people are coming into contact with predictions of a GNN, on a day-to-day basis. As such, it is likely a valuable pursuit to study GNNs, even without aiming to directly contribute to their development.

Beyond this, it is likely that the very *cognition* processes driving our reasoning and decision-making are, in some sense, *graph-structured*. That is, paraphrasing a quote from Forrester (1971), nobody really imagines in their head all the information known to them; rather, they imagine only selected *concepts*, and *relationships* between them, and use those to represent the real system. If we subscribe to this interpretation of cognition, it is quite unlikely that we will be able to build a generally intelligent system without some component relying on graph representation learning. Note that this finding does not clash with the fact that many recent skillful ML systems are based on the Transformer architecture (Vaswani et al., 2017)—as we will uncover in this review, Transformers are themselves a special case of GNNs.

2 The fundamentals: Permutation equivariance and invariance

In the previous section, we saw *why* it is a good idea to study data that lives on graphs. Now we will see *how to learn* useful functions over graph-structured data. The exposition largely follows Bronstein et al. (2021).

With graph-structured inputs, we typically assume a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; that is, we have a set of *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, which specifies pairs of nodes in $\mathcal{V}$ that are connected.

As we are interested in representation learning over the nodes, we attach to each node $u \in \mathcal{V}$ a feature vector, $\mathbf{x}_u \in \mathbb{R}^k$. The main way in which this data is *presented* to a machine learning model is in the form of a *node feature matrix*. That is, a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times k}$ is prepared by stacking these features:

$$\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{V}|} \right]^\top \tag{1}$$

that is, the ith row of $\mathbf{X}$ corresponds to $\mathbf{x}_i$.

There are many ways to represent $\mathcal{E}$; since our context is one of linear algebra, we will use the *adjacency matrix*, $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$:

$$a_{uv} = \begin{cases} 1 & (u, v) \in \mathcal{E} \\ 0 & (u, v) \notin \mathcal{E} \end{cases} \tag{2}$$

Note that it is often possible, especially in biochemical inputs, that we want to attach more information to the edges (such as distance scalars, or even entire feature vectors). I deliberately do not consider such cases to retain clarity—the conclusions we make would be the same in those cases.

However, the very act of using the above representations imposes a *node ordering*, and is therefore an arbitrary choice which does not align with the nodes and edges being unordered! Hence, we need to make sure that permuting the nodes and edges ($\mathbf{PAP}^\top$, for a permutation matrix $\mathbf{P}$), does not change the outputs. We recover the following rules a GNN must satisfy:

$$f(\mathbf{PX}, \mathbf{PAP}^\top) = f(\mathbf{X}, \mathbf{A}) \qquad \text{(Invariance)} \tag{3}$$
$$\mathbf{F}(\mathbf{PX}, \mathbf{PAP}^\top) = \mathbf{PF}(\mathbf{X}, \mathbf{A}) \qquad \text{(Equivariance)} \tag{4}$$

Here we assumed for simplicity that the functions f, $\mathbf{F}$ do not change the adjacency matrix, so we assume they only return graph or node-level outputs.

Further, the graph's edges allow for a *locality* constraint in these functions. Much like how a CNN operates over a small neighbourhood of each pixel of an image, a GNN can operate over a neighbourhood of a node. One standard way to define this neighbourhood, $\mathcal{N}_u$, is as follows:

$$\mathcal{N}_u = \{v \mid (u, v) \in \mathcal{E} \vee (v, u) \in \mathcal{E}\} \tag{5}$$

Accordingly, we can define the multiset of all neighbourhood features, $\mathbf{X}_{\mathcal{N}_u}$:

$$\mathbf{X}_{\mathcal{N}_u} = \{\!\{\mathbf{x}_v \mid v \in \mathcal{N}_u\}\!\} \tag{6}$$

And our local function, ϕ, can take into account the neighbourhood; that is:

$$\mathbf{h}_u = \phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u}) \qquad \mathbf{F}(\mathbf{X}) = \left[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{|\mathcal{V}|}\right]^\top \tag{7}$$

Through simple linear algebra manipulation, it is possible to show that if ϕ is permutation invariant in $\mathbf{X}_{\mathcal{N}_u}$, then $\mathbf{F}$ will be permutation equivariant. The remaining question is, how do we define ϕ?

3 Graph Neural Networks

Needless to say, defining ϕ is one of the most active areas of machine learning research today. Depending on the literature context, it may be referred to as either "diffusion", "propagation", or "message passing". As claimed by Bronstein et al. (2021), most of them can be classified into one of three spatial flavours:

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{vu}\psi(\mathbf{x}_v)\right) \qquad \text{(Convolutional)} \tag{8}$$

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v)\psi(\mathbf{x}_v)\right) \qquad \text{(Attentional)} \tag{9}$$

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v)\right) \qquad \text{(Message-passing)} \tag{10}$$

where ψ and ϕ are neural networks—e.g. $\psi(\mathbf{x}) = \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b})$, and $\bigoplus$ is any permutation-invariant aggregator, such as $\sum$, averaging, or max. The expressive power of the GNN progressively increases going from Equation 8 to 10, at the cost of interpretability, scalability, or learning stability. For most tasks, a careful tradeoff is needed when choosing the right flavour.

4

This review does not attempt to be a comprehensive overview of specific GNN layers. That being said: representative *convolutional* GNNs include the Chebyshev network (Defferrard et al., 2016, ChebyNet), graph convolutional network (Kipf and Welling, 2017, GCN) and the simplified graph convolution (Wu et al., 2019, SGC); representative *attentional* GNNs include the mixture model CNN (Monti et al., 2017, MoNet), graph attention network (Veličković et al., 2018, GAT) and its recent "v2" variant (Brody et al., 2022, GATv2); and representative *message-passing* GNNs include interaction networks (Battaglia et al., 2016, IN), message passing neural networks (Gilmer et al., 2017, MPNN) and graph networks (Battaglia et al., 2018, GN).

Given such a GNN layer, we can learn (m)any interesting tasks over a graph, by appropriately combining $\mathbf{h}_u$. I exemplify the three principal such tasks, grounded in biological examples:

Node classification. If the aim is to predict targets for each node $u \in \mathcal{V}$, then our output is equivariant, and we can learn a shared classifier directly on $\mathbf{h}_u$. A canonical example of this is classifying protein functions (e.g. using gene ontology data (Zitnik and Leskovec, 2017)) in a given protein-protein interaction network, as first done by GraphSAGE (Hamilton et al., 2017).

Graph classification. If we want to predict targets for the entire graph, then we want an invariant output, hence need to first *reduce* all the $\mathbf{h}_u$ into a common representation, e.g. by performing $\bigoplus_{u \in \mathcal{V}} \mathbf{h}_u$, then learning a classifier over the resulting flat vector. A canonical example is classifying molecules for their quantum-chemical properties (Gilmer et al., 2017), estimating pharmacological properties like toxicity or solubility (Duvenaud et al., 2015; Xiong et al., 2019; Jiang et al., 2021) or virtual drug screening (Stokes et al., 2020).

Link prediction. Lastly, we may be interested in predicting properties of *edges* (u, v), or even predicting whether an edge *exists*; giving rise to the name "link prediction". In this case, a classifier can be learnt over the concatenation of features $\mathbf{h}_u \| \mathbf{h}_v$, along with any given edge-level features. Canonical tasks include predicting links between drugs and diseases—drug repurposing (Morselli Gysi et al., 2021), drugs and targets—binding affinity prediction (Lim et al., 2019; Jiang et al., 2020), or drugs and drugs—predicting adverse side-effects from polypharmacy (Zitnik et al., 2018; Deac et al., 2019).

It is possible to use the building blocks from the principal tasks above to go beyond classifying the entities given by the input graph, and have systems that *produce* novel molecules (Mercado et al., 2021) or even perform *retrosynthesis*—the estimation of which reactions to utilise to synthesise given molecules (Somnath et al., 2021; Liu et al., 2022).

A natural question arises, following similar discussions over sets (Zaheer et al.,

2017; Wagstaff et al., 2019): Do GNNs, as given by Equation 10, represent *all* of the valid permutation-equivariant functions over graphs? Opinions are divided. Key results in previous years seem to indicate that such models are fundamentally limited in terms of problems they can solve (Xu et al., 2019; Morris et al., 2019). However, most, if not all, of the proposals for addressing those limitations are still expressible using the pairwise message passing formalism of Equation 10; the main requirement is to carefully modify the *graph* over which the equation is applied (Veličković, 2022). To supplement this further, Loukas (2020) showed that, under proper initial features, sufficient depth-width product ($\#$layers $\times$ dim $\mathbf{h}_u$), and correct choices of ψ and ϕ, GNNs in Equation 10 are *Turing universal*—likely to be able to simulate *any* computation which any computer can perform over such inputs.

All points considered, it is the author's opinion that the formalism in this section is likely all we need to build powerful GNNs—although, of course, different perspectives may benefit different problems, and existence of a powerful GNN does not mean it is easy to find using stochastic gradient descent.

4 GNNs without a graph: Deep Sets and Transformers

Throughout the prior section, we have made a seemingly innocent assumption: that we are *given* an input graph (through $\mathbf{A}$). However, very often, not only will there not be a clear choice of $\mathbf{A}$, but we may not have any prior belief on what $\mathbf{A}$ even is. Further, even if a ground-truth $\mathbf{A}$ is given *without noise*, it may not be the optimal *computation graph*: that is, passing messages over it may be problematic—for example, due to bottlenecks (Alon and Yahav, 2021). As such, it is generally a useful pursuit to study GNNs that are capable of modulating the input graph structure.

Accordingly, let us assume we only have a node feature matrix $\mathbf{X}$, but no adjacency. One simple option is the "pessimistic" one: assume there are no edges at all, i.e. $\mathbf{A} = \mathbf{I}$, or $\mathcal{N}_u = \{u\}$. Under such an assumption, Equations 8–10 *all* reduce to $\mathbf{h}_u = \phi(\mathbf{x}_u)$, yielding the Deep Sets model (Zaheer et al., 2017). Therefore, no power from graph-based modelling is exploited here.

The converse option (the "lazy" one) is to, instead, assume a *fully-connected* graph; that is $\mathbf{A} = \mathbf{1}\mathbf{1}^\top$, or $\mathcal{N}_u = \mathcal{V}$. This then gives the GNN the full potential to exploit any edges deemed suitable, and is a very popular choice for smaller numbers of nodes. It can be shown that convolutional GNNs (Equation 8) would still reduce to Deep Sets in this case, which motivates the use of a stronger GNN. The next model

in the hierarchy, attentional GNNs (Equation 9), reduce to the following equation:

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{V}} a(\mathbf{x}_u, \mathbf{x}_v)\psi(\mathbf{x}_v)\right) \tag{11}$$

which is essentially the forward pass of a Transformer (Vaswani et al., 2017). To reverse-engineer why Transformers appear here, let us consider the NLP perspective. Namely, words in a sentence *interact* (e.g. subject-object, adverb-verb). Further, these interactions are not trivial, and certainly not *sequential*—that is, words can interact even if they are many sentences apart[1]. Hence, we may want to use a *graph* between them. But what *is* this graph? Not even annotators tend to agree, and the optimal graph may well be task-dependant. In such a setting, a common assumption is to use a complete graph, and let the network infer relations by itself—at this point, the Transformer is all but rederived. For an in-depth rederivation, see Joshi (2020).

Another reason why Transformers have become such a dominant GNN variant is the fact that using a fully connected graph structure allows to express all model computations using *dense matrix products*, and hence their computations align very well with current prevalent accelerators (GPUs and TPUs). Further, they have a more favourable storage complexity than the message passing variant (Equation 10). Accordingly, Transformers can be seen as GNNs that are currently winning the hardware lottery (Hooker, 2021)!

Before closing this section, it is worth noting a *third* option to learning a GNN without an input graph: to *infer* a graph structure to be used as edges for a GNN. This is an emerging area known as *latent graph inference*. It is typically quite challenging, since edge selection is a non-differentiable operation, and various paradigms have been proposed in recent years to overcome this challenge: nonparametric (Wang et al., 2019; Deac et al., 2022), supervised (Veličković et al., 2020), variational (Kipf et al., 2018), reinforcement (Kazi et al., 2022) and self-supervised learning (Fatemi et al., 2021).

5 GNNs beyond permutation equivariance: Geometric Graphs

To conclude our discussion, we revisit another assumption: we have assumed our graphs to be a discrete, unordered, collection of nodes and edges—hence, only susceptible to permutation symmetries. But in many cases, this is not the entire story!

---

[1]This insight may also partly explain why RNNs or CNNs have been seen as suboptimal language models: they implicitly assume only neighbouring words directly interact.

The graph, in fact, may often be endowed with some spatial *geometry*, which will be very useful to exploit. Molecules, and their three-dimensional conformer structure, are a classical example of this.

In general, we will assume our inputs are *geometric graphs*: nodes are endowed with both *features*, $\mathbf{f}_u$, and *coordinates*, $\mathbf{x}_u \in \mathbb{R}^3$. We may be interested in designing a model that is equivariant not only to permutations, but also 3D rotations, translations and reflections (the Euclidean group, E(3)).

An E(3)-equivariant message passing layer transforms the coordinates and features, and yields updated features $\mathbf{f}'_u$ and coordinates $\mathbf{x}'_u$. There exist many GNN layers that obey E(n) equivariance, and one particularly elegant solution was proposed by Satorras et al. (2021):

$$\mathbf{f}'_u = \phi \left(\mathbf{f}_u, \bigoplus_{v \in \mathcal{N}_u} \psi_{\mathrm{f}} \left(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2 \right) \right) \tag{12}$$

$$\mathbf{x}'_u = \mathbf{x}_u + \sum_{v \neq u} (\mathbf{x}_u - \mathbf{x}_v) \psi_{\mathrm{x}} \left(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2 \right) \tag{13}$$

The key insight behind this model is that rotating, translating or reflecting coordinates does not change their distances $\|\mathbf{x}_u - \mathbf{x}_v\|^2$, i.e., such operations are *isometries*. Hence, if we roto-translate all nodes as $\mathbf{x}_u \leftarrow \mathbf{R}\mathbf{x}_u + \mathbf{b}$, the output features $\mathbf{f}'_u$ remain unchanged, while the output coordinates transform accordingly: $\mathbf{x}'_u \leftarrow \mathbf{R}\mathbf{x}'_u + \mathbf{b}$.

While indeed highly elegant, a model like this hides a caveat: it only works over *scalar* features $\mathbf{f}_u$. If our model needs to support any kind of *vector* input (e.g. *forces* between atoms), the model in Equation 12 would not suffice, because the vectors would need to appropriately rotate with $\mathbf{R}$. Satorras et al. (2021) do propose a variant that allows for explicitly updating vector features, $\mathbf{v}_u$:

$$\mathbf{v}'_u = \phi_{\mathrm{v}}(\mathbf{h}_u)\mathbf{v}_u + C \sum_{v \neq u} (\mathbf{x}_u - \mathbf{x}_v) \phi_{\mathrm{x}} \left(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2 \right), \quad \mathbf{x}'_u = \mathbf{x}_u + \mathbf{v}'_u \tag{14}$$

But these issues will continue to arise, as features get "tensored up". Hence, in such circumstances, it might be useful to instead characterise a generic equation that supports *all* possible roto-translation equivariant models, and then learning its parameters. Such an analysis was done in Tensor Field Networks (Thomas et al., 2018) for point clouds, and then extended to SE(3)-Transformers for general graphs (Fuchs et al., 2020).

Perhaps a fitting conclusion of this survey is a simple realisation: having showed how both Transformers and geometric equivariance constraints play a part within the context of GNNs, we now have all of the key building blocks to define some

of the most famous geometric GNN architectures in the wild, such as AlphaFold 2 (Jumper et al., 2021), but also similar protein-related papers which made headlines in both Nature Methods (Gainza et al., 2020, MaSIF) and Nature Machine Intelligence (Méndez-Lucio et al., 2021). It seems that protein folding, protein design, and protein binding prediction (Stärk et al., 2022) all appear to be an extremely potent area of attack for geometric GNNs; just one of many solid reasons why the field of structural biology would benefit from these recent developments (Bouatta et al., 2021).

References

Alon, U., Yahav, E., 2021. On the bottleneck of graph neural networks and its practical implications, in: International Conference on Learning Representations. URL: `https://openreview.net/forum?id=i80OPhOCVH2`.

Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al., 2016. Interaction networks for learning about objects, relations and physics. Advances in neural information processing systems 29.

Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al., 2018. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 .

Blundell, C., Buesing, L., Davies, A., Veličković, P., Williamson, G., 2021. Towards combinatorial invariance for kazhdan-lusztig polynomials. arXiv preprint arXiv:2111.15161 .

Bouatta, N., Sorger, P., AlQuraishi, M., 2021. Protein structure prediction by alphafold2: are attention and symmetries all you need? Acta Crystallographica Section D: Structural Biology 77, 982–991.

Brody, S., Alon, U., Yahav, E., 2022. How attentive are graph attention networks?, in: International Conference on Learning Representations. URL: `https://openreview.net/forum?id=F72ximsx7C1`.

Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P., 2021. **Geometric deep learning: Grids, groups, graphs, geodesics, and gauges**. arXiv preprint arXiv:2104.13478 (**) Generalises the concepts in graph neural networks to derive most of deep learning in use today, satisfying specific symmetry groups over input domains.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., et al., 2021. **Advancing mathematics by guiding human intuition with AI**. Nature 600, 70–74. (*) Used GNNs to uncover the structure of objects in pure mathematics, leading to top-tier conjectures and theorems in two distinct areas of mathematics.

Deac, A., Huang, Y.H., Veličković, P., Liò, P., Tang, J., 2019. Drug-drug adverse effect prediction with graph co-attention. arXiv preprint arXiv:1905.00534 .

Deac, A., Lackenby, M., Veličković, P., 2022. Expander graph propagation, in: The First Learning on Graphs Conference. URL: `https://openreview.net/forum?id=IKevTLt3rT`.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 29.

Derrow-Pinion, A., She, J., Wong, D., Lange, O., Hester, T., Perez, L., Nunkesser, M., Lee, S., Guo, X., Wiltshire, B., et al., 2021. **Eta prediction with graph neural networks in google maps**, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3767–3776. (*) Deployed GNNs to serve travel time predictions in Google Maps.

Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P., 2015. Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems 28.

Fatemi, B., El Asri, L., Kazemi, S.M., 2021. Slaps: Self-supervision improves structure learning for graph neural networks. Advances in Neural Information Processing Systems 34, 22667–22681.

Forrester, J.W., 1971. Counterintuitive behavior of social systems. Theory and decision 2, 109–140.

Fuchs, F., Worrall, D., Fischer, V., Welling, M., 2020. Se (3)-transformers: 3d roto-translation equivariant attention networks. Advances in Neural Information Processing Systems 33, 1970–1981.

Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., Correia, B., 2020. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods 17, 184–192.

Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry, in: International conference on machine learning, PMLR. pp. 1263–1272.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Advances in neural information processing systems 30.

Hao, J., Zhao, T., Li, J., Dong, X.L., Faloutsos, C., Sun, Y., Wang, W., 2020. **P-companion: A principled framework for diversified complementary product recommendation**, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2517–2524. **(*)** Deployed GNNs in important parts of Amazon's product recommendation system.

Hooker, S., 2021. The hardware lottery. Communications of the ACM 64, 58–65.

Jiang, D., Wu, Z., Hsieh, C.Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., Hou, T., 2021. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. Journal of cheminformatics 13, 1–23.

Jiang, M., Li, Z., Zhang, S., Wang, S., Wang, X., Yuan, Q., Wei, Z., 2020. Drug–target affinity prediction using graph neural network and contact maps. RSC advances 10, 20701–20712.

Joshi, C., 2020. **Transformers are graph neural networks**. The Gradient , 5 **(**)** Demonstrates that Transformers are attentional GNNs.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al., 2021. **Highly accurate protein structure prediction with AlphaFold**. Nature 596, 583–589. **(*)** The AlphaFold 2 model, demonstrating highly accurate predictions of protein structure given their amino acid sequence.

Kazi, A., Cosmo, L., Ahmadi, S.A., Navab, N., Bronstein, M., 2022. **Differentiable graph module (dgm) for graph convolutional networks**. IEEE Transactions on Pattern Analysis and Machine Intelligence **(*)** Uses a combination of GNNs and RL to discover latent graph structures in point clouds and biomedical data.

Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R., 2018. Neural relational inference for interacting systems, in: International Conference on Machine Learning, PMLR. pp. 2688–2697.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=SJU4ayYgl.

Lim, J., Ryu, S., Park, K., Choe, Y.J., Ham, J., Kim, W.Y., 2019. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. Journal of chemical information and modeling 59, 3981–3988.

Liu, C.H., Korablyov, M., Jastrzebski, S., Włodarczyk-Pruszynski, P., Bengio, Y., Segler, M., 2022. Retrognn: Fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software. Journal of Chemical Information and Modeling 62, 2293–2300.

Loukas, A., 2020. **What graph neural networks cannot learn: depth vs width**, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=B1l2bp4YwS. (*) Demonstrates that MPNNs are Turing universal, under reasonable assumptions on their input and parametrisation.

Méndez-Lucio, O., Ahmad, M., del Rio-Chanona, E.A., Wegner, J.K., 2021. A geometric deep learning approach to predict binding conformations of bioactive molecules. Nature Machine Intelligence 3, 1033–1039.

Mercado, R., Rastemo, T., Lindelöf, E., Klambauer, G., Engkvist, O., Chen, H., Bjerrum, E.J., 2021. Graph networks for molecular design. Machine Learning: Science and Technology 2, 025023.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J.W., Songhori, E., Wang, S., Lee, Y.J., Johnson, E., Pathak, O., Nazi, A., et al., 2021. **A graph placement methodology for fast chip design**. Nature 594, 207–212. (*) Used GNNs to optimise chip placement to superhuman level, with GNN-powered designs making it into the TPUv5.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model cnns, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5115–5124.

Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M., 2019. Weisfeiler and leman go neural: Higher-order graph neural networks, in: Proceedings of the AAAI conference on artificial intelligence, pp. 4602–4609.

Morselli Gysi, D., Do Valle, Í., Zitnik, M., Ameli, A., Gan, X., Varol, O., Ghiassian, S.D., Patten, J., Davey, R.A., Loscalzo, J., et al., 2021. Network medicine framework for identifying drug-repurposing opportunities for covid-19. Proceedings of the National Academy of Sciences 118, e2025581118.

Satorras, V.G., Hoogeboom, E., Welling, M., 2021. **E (n) equivariant graph neural networks**, in: International conference on machine learning, PMLR. pp. 9323–9332. **(**)** An elegant, performant E(n)-equivariant GNN.

Somnath, V.R., Bunne, C., Coley, C., Krause, A., Barzilay, R., 2021. Learning graph models for retrosynthesis prediction. Advances in Neural Information Processing Systems 34, 9405–9415.

Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., Jaakkola, T., 2022. Equibind: Geometric deep learning for drug binding structure prediction, in: International Conference on Machine Learning, PMLR. pp. 20503–20521.

Stokes, J.M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N.M., MacNair, C.R., French, S., Carfrae, L.A., Bloom-Ackermann, Z., et al., 2020. **A deep learning approach to antibiotic discovery**. Cell 180, 688–702. **(**)** Discovered a novel, potent antibiotic (Halicin) using a GNN-based virtual drug screening system.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., Riley, P., 2018. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv preprint arXiv:1802.08219 .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Veličković, P., 2022. **Message passing all the way up**, in: ICLR 2022 Workshop on Geometrical and Topological Representation Learning. URL: `https://openreview.net/forum?id=Bc8GiEZkTe5`. **(*)** Argues for one-hop MPNNs being sufficient to represent any GNN of interest, if we allow for input graph modulation.

Veličković, P., Buesing, L., Overlan, M., Pascanu, R., Vinyals, O., Blundell, C., 2020. Pointer graph networks. Advances in Neural Information Processing Systems 33, 2232–2244.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: International Conference on Learning Representations. URL: `https://openreview.net/forum?id=rJXMpikCZ`.

Wagstaff, E., Fuchs, F., Engelcke, M., Posner, I., Osborne, M.A., 2019. On the limitations of representing functions on sets, in: International Conference on Machine Learning, PMLR. pp. 6487–6494.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) 38, 1–12.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K., 2019. Simplifying graph convolutional networks, in: International conference on machine learning, PMLR. pp. 6861–6871.

Xiong, Z., Wang, D., Liu, X., Zhong, F., Wan, X., Li, X., Li, Z., Luo, X., Chen, K., Jiang, H., et al., 2019. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. Journal of medicinal chemistry 63, 8749–8760.

Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2019. How powerful are graph neural networks?, in: International Conference on Learning Representations. URL: `https://openreview.net/forum?id=ryGs6iA5Km`.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J., 2018. Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 974–983.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J., 2017. Deep sets. Advances in neural information processing systems 30.

Zitnik, M., Agrawal, M., Leskovec, J., 2018. Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 34, i457–i466.

Zitnik, M., Leskovec, J., 2017. Predicting multicellular function through multi-layer tissue networks. Bioinformatics 33, i190–i198.