TOWARDS LEARNING KNOT INVARIANTS USING GRAPH NEURAL NETWORKS

William Koch

ABSTRACT

So far, very little research has been done on the applications of machine learning to the topological field of knot theory. To the best of our knowledge, we are the first to investigate directly leveraging the topological structure of knots for learning tasks to predict certain properties.

We show that graph neural networks, in particular graph isomorphism networks, provide better accuracy in predicting the symmetry of prime knots, indicating that the structure of knots can indeed be leveraged. While the performance of models is still not sufficient for useful applications in research in mathematics, we hope that this study inspires further research at the intersection of knot theory and geometric learning methods.

1 INTRODUCTION

The purpose of this study is to investigate whether graph neural networks can be a suitable approach to predicting knot invariants by converting knots to graph representations and thus enabling the network to indirectly learn on the topological structure of the knot. This can have several interesting applications in knot-theory, such as addressing the open problem of checking whether two knots are equivalent Lackenby (2017). The complexity of this problem is unknown, but if graph-based approaches turn out to be sufficiently expressive, we could develop efficient heuristics. Not only could this also lead to a better understanding of the topology, but it may also provide for more efficient means to expand the table of prime knots.

Means and Outline. To answer this question, we first provide a theoretical intuition of the topology. This serves as a basis for the construction of our graph dataset, which is based on a table of known prime knots with pre-computed properties. We show that our graph representation uniquely describes a knot.

In two experiments, we then experimentally verify that the graph representation is indeed related to knot properties. First, we want to test whether the relational structure alone is actually sufficient for this task. This is because the graph corresponding to a planar diagram may actually represent different knots. We consider alternative features to ensure that the corresponding knot is unique, and evaluate the model performance differences. In a second experiment, we then test different architectures against a standard MLP to confirm the hypothesis.

Related Work. Some of the more recent research in knot theory uses neural networks to understand the correlation between the Jones and specialized Khovanov polynomials, and the knots' s and g-invariant (Rasmussen invariant and slice genus) Craven et al. (2023). Similarly, it has been shown that given several known knot properties and one-hot encoded braid representations, neural networks can accurately predict whether a knot is quasipositive or quasinegative Hughes (2016). However, to the best of our knowledge, no previous research has been done on directly learning on graph representations of knots. However, in related fields, there is more research on graph neural network applications. Especially insights from molecular chemistry can be useful, where chirality plays an important role and research has already been conducted Adams et al. (2021).

Contributions. To the best of our knowledge, this is the first study attempting to apply graph neural networks to knot theory. Our contributions are therefore twofold. Firstly, we construct a dataset of graph representations of knots from the prime knot database KnotInfo Livingston & Moore (Current Year), and secondly, we show that graph neural networks can predict graph invariants with greater accuracy than MLPs, which proves that certain knot invariants (in this case: symmetry type) can be

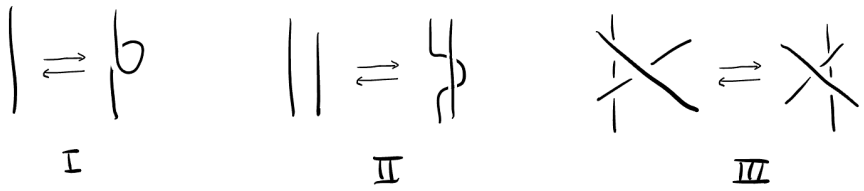

Figure 1: The three Reidemeister moves under which two knots remain equivalent.

learned from the topological structure alone, rather than from other pre-computed knot invariants as previous literature has done.

2 THEORETICAL BACKGROUND

We briefly summarize some of the relevant theoretical background on knot theory from Lickorish (1997). Knot theory is a branch of topology that intuitively studies how connected one-dimensional strings can be arranged in three-dimensional space. We rephrase definition 1.1 from Lickorish (1997) to be specific to knots, as we do not consider the more general case of links with more than one component:

Definition. *A knot is a subset of S^3, or R^3, that consists of a single, piecewise linear, simple closed curve.*

Two knots are said to be equivalent if there exists an orientation-preserving piecewise linear homeomorphism that maps both knots to the same value. A knot invariant is a quantity which remains constant under any two equivalent knots Lickorish (1997).

Reidemeister showed that any two equivalent knots can be related through a homeomorphism consisting of only three moves Reidemeister (1927), the so called Reidemeister moves (see figure 1).

Any knot can be represented by a so-called planar diagram (PD). Here, each edge of the plane graph representation of the knot is labeled by a number. The PD representation is a list of crossings of the knot, where each crossing is identified by four numbers which correspond to the connecting edges. The first denotes the incoming lower edge of the crossing, and the others are the remaining edges, in a counter-clockwise order noa.

According to the Jordan Curve Theorem, in the plane graph representation (step 1 in figure 3) there exists exactly one coloring where the unbounded face is colored white and the remaining surfaces are colored black, i.e. the planar graph is 2-colorable. The black faces describe the Seifert surface of the knot.

We finally distinguish between two crossings in this colored planar graph, left- and right-handed crossings (see figure 2). Distinguishing between the two is important, as it may affect the symmetry of the knot, or the knot all together.

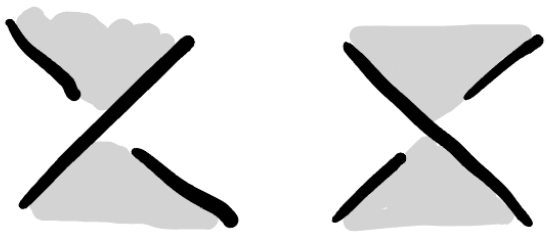

Figure 2: From left to right: left-handed and right-handed crossing.

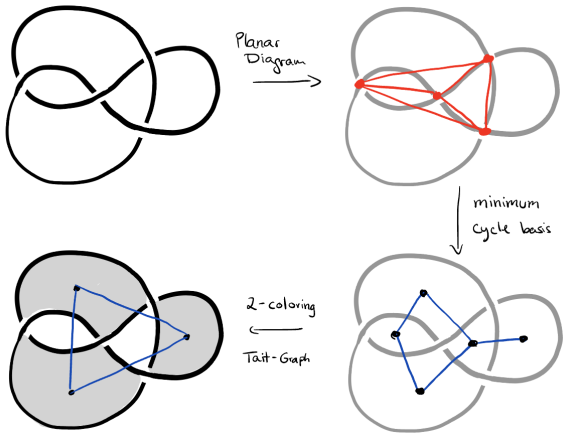

Figure 3: Coloring a knot

3 Preparing the Data: Graph Representation of Knots

We use the KnotInfo database Livingston & Moore (Current Year) for a table of all prime knots up to and including 12 crossings. As a basis, we use the plane graph generated from the planar diagram representation (step 1 and 2 in figure 3). While we will test the model performance directly on this graph structure, this may not be expressive enough, since the plane graph does not have any information about the different crossing types. We point out that different crossing types may result in different knots, so we take further steps to make the representation more expressive.

In the next step, from the plane graph, we construct a second graph where each node corresponds to a face in the planar graph of the node by computing the minimum cycle basis of the plane graph. We then 2-color this graph, and drop all nodes on white faces (steps 3 and 4 in figure 3).

Now that we have the coloring of the faces, we return to our original basic plane graph. As each node represents a crossing, we refer to the planar diagram representation described in section 2 (different from the plane graph). Knowing the incoming lower edge, as well as the next edge in the counter-clockwise order, we can check whether the two edges span a black face in the colored plane graph. This tells us whether a node is a left-handed or a right-handed crossing. If the node is a left-handed crossing, we assign it the feature +1, and if it is a right-handed crossing, -1.

The provided dataset class performs these conversions during preprocessing[1].

Limitations. In this representation, some edges might have duplicate edges in the corresponding knot; however, they can only be represented as one edge. In fact, the graph is unable to distinguish between a single and a double edge[2]. However, intuitively, this should not be an issue in most cases - each crossing requires two incoming and two outgoing edges, enforced by the planar diagram representation. The handedness of the crossing should, in most cases, provide the information necessary to infer the double edges.

Furthermore, we point out that the Reidmeister moves can introduce bottlenecks, especially the second one. Here, two edges in a graph are equivalent to two nodes connected by a single edge, with each two additional outgoing edges. This may limit information flow if the knot is not simplified. However, all knots in our experiment are simple prime knots.

The resulting database consists of 2,890 graphs, with 5-12 nodes each.

[1]We did not find an algorithm to compute the handedness of the crossings for our use case, so we inferred these steps from the general properties and verified it on several knots

[2]Note that triple edges are impossible, as it would involve the crossing of three edges at a single point, which is prohibited by the planar diagram representation on which our graph is based.

4 EXPERIMENTS

Setup. Using the graph representation of knots outlined in section 3, we aim to predict the symmetry type of the knot, i.e. determine whether a given prime knot is either reversible or chiral. While our database lists a total of six symmetry types, we only consider these two, as the others are strongly underrepresented; each contribute less than 1.5%. 54% of knots in the filtered dataset are reversible, whereas 46% are chiral. The resulting dataset is rather homophilious with respect to the crossing types which act as input features, with a node homophily of 0.71. This is unsurprising, given that most prime knots are alternating Lickorish (1997), which implies that they admit crossings that are often entirely left- or right-handed.

We show that graph neural networks outperform neural networks in predicting the symmetry of knots, and perform two ablation studies. Firstly, we analyze the importance of the input features, by first setting all node features either to 1, and then to +1 for left-handed crossings and -1 for right-handed crossings. Secondly, we investigate the importance of the number of layers for large knots, i.e. the importance of full information propagation.

Chosen models. We evaluate four different architectures - Graph Convolution Networks (GCN) Kipf & Welling (2017), Graph Attention Networks (GAT) Veličković et al. (2022) and Graph Isomorphism Networks (GIN) Xu et al. (2019). Since we have up to twelve crossings in our dataset, the maximum distance between two nodes in a graph is six[3]. For the information to pass through the entire knot, we therefore consider architectures with at least six layers. Due to the higher layer count, we use Jumping-Knowledge Xu et al. (2018) aggregations in all models to address over-smoothing, concatenating the representations from every layer. Each model is followed by a global mean pooling layer and a 2-layer MLP for the graph-level classification task.

Hyperparameters. All models are trained on an Nvidia RTX3090 GPU, and each model is initially fine-tuned over 50 runs at 75 epochs each (no improvements beyond that observed) using the Bayes method, optimizing for the validation accuracy. During fine-tuning and all other stages, we use five random training, validation and test splits and report the mean and standard deviation of the test accuracy[4]. We use a scheduled learning rate, starting at 1e-3, patience of 5, decaying factor of 0.9 and a minimum learning rate of 1e-5. The size of the hidden layers d_{hidden} is chosen from [16, 32, 64, 128, 256], the number of layers as one of [2, 4, 8, 12] and the normalization is one of batch or layer normalization.

Table 1: Hyperparameter tuning results

Architecture	d_{hidden}	n_{layers}	Norm
GIN	32	12	BatchNorm
GCN	128	6	LayerNorm
GAT	64	4	LayerNorm
MLP	128	6	BatchNorm

Results. We observe that encoding the crossing type into the node features improves the performance of the Graph Isomorphism Network by approximately 4%. Surprisingly, however, it also leads to an improvement for the basic MLP architecture. In the latter case, we suspect that the symmetry distribution of alternating and non-alternating knots changes, which the MLP could pick up on.

Encoding the crossing types into the node features, we compare the performance of additional networks against the MLP as a baseline. We observe that the MLP architecture achieves an accuracy close to 54%, which is the occurrence of the most frequent symmetry type and therefore not much better than an educated guess.

[3] As all considered knots are tied, the worst case occurs when the graph representation of the knot is cyclical.

[4] For each split, we report the test accuracies of the epochs with the best validation accuracies and aggregate them.

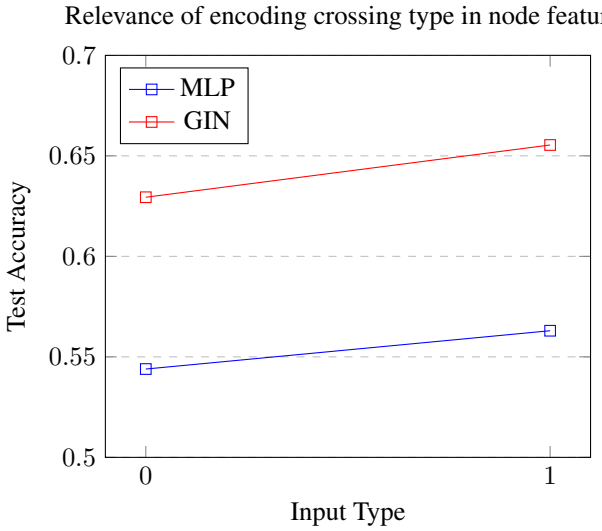

Figure 4: For input type 0, we set all node features to 1, whereas for input type 1, we provide information about the crossing. Both MLP and GIN benefit from information on the crossing type.

On the other hand, all graph-based models outperform the MLP, the Graph Isomorphism Network in particular. Learning the aggregation function appears to be useful for predicting the symmetry of knots.

Table 2: Predicting the symmetry type of knots with up to 12 crossings

Architecture	Test Accuracy
GIN	$65.54 \pm 1.91\%$
GCN	$58.79 \pm 1.85\%$
GAT	$58.27 \pm 1.26\%$
MLP	$56.30 \pm 1.74\%$

5 Conclusion

In this work, we have shown that through an appropriate graph representation of prime knots, the topological structure can be used to improve predictions of certain properties, such as the symmetry type of knots.

Limitations. As pointed out in section 3, the graph representation of the knots results in potential bottlenecks during training caused by unnecessary crossings. While this is not the case for our dataset, as the prime knots have a minimal number of crossings, this might be a limitation for random knots.

In this study, we have only considered prime knots. It will, in particular, be relevant to investigate whether graph-based approaches are able to express the Reidemeister moves on more general (non-prime) knots, which describe the rules under which knots can be considered equivalent. There may be more suitable architectures or graph-representations of knots to express these equivalence relations. Then we can make more conclusive arguments about the ability of graph neural networks to reflect the topological structure of knots in general.

Outlook. If further studies prove successful, graph neural networks could be used to assist with the fundamental problem of checking whether two knots are equivalent. This is an open problem, with unknown complexity. Similarly, this could open doors to graph-generation tasks, such as generating

likely prime-knot candidates. These applications could assist researchers in expanding the table of prime knots to knots with more than 19 crossings Burton (2020).

REFERENCES

Planar Diagrams - Knot Atlas. URL http://katlas.org/wiki/Planar_Diagrams.

Keir Adams, Lagnajit Pattanaik, and Connor W. Coley. Learning 3D Representations of Molecular Chirality with Invariance to Bond Rotations, October 2021. URL http://arxiv.org/abs/2110.04383. arXiv:2110.04383 [cs].

Benjamin A. Burton. The Next 350 Million Knots. In Sergio Cabello and Danny Z. Chen (eds.), *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 25:1–25:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-143-6. doi: 10.4230/LIPIcs.SoCG.2020.25. URL https://drops.dagstuhl.de/opus/volltexte/2020/12183. ISSN: 1868-8969.

Jessica Craven, Mark Hughes, Vishnu Jejjala, and Arjun Kar. Learning knot invariants across dimensions. *SciPost Physics*, 14(2):021, February 2023. ISSN 2542-4653. doi: 10.21468/SciPostPhys.14.2.021. URL http://arxiv.org/abs/2112.00016. arXiv:2112.00016 [hep-th].

Mark C. Hughes. A neural network approach to predicting and computing knot invariants, October 2016. URL http://arxiv.org/abs/1610.05744. arXiv:1610.05744 [math].

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. URL http://arxiv.org/abs/1609.02907. arXiv:1609.02907 [cs, stat].

Marc Lackenby. *Elementary Knot Theory*, volume 1. Oxford University Press, May 2017. doi: 10.1093/oso/9780198784913.003.0002. URL https://academic.oup.com/book/26786/chapter/195720771.

W. B. Raymond Lickorish. A Beginning for Knot Theory. In W. B. Raymond Lickorish (ed.), *An Introduction to Knot Theory*, Graduate Texts in Mathematics, pp. 1–14. Springer, New York, NY, 1997. ISBN 978-1-4612-0691-0. doi: 10.1007/978-1-4612-0691-0_1. URL https://doi.org/10.1007/978-1-4612-0691-0_1.

Charles Livingston and Allison H. Moore. Knotinfo: Table of knot invariants. URL: knotinfo.math.indiana.edu, Current Month Current Year.

Kurt Reidemeister. Elementare Begründung der Knotentheorie. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 5(1):24–32, December 1927. ISSN 1865-8784. doi: 10.1007/BF02952507. URL https://doi.org/10.1007/BF02952507.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. March 2022. URL https://openreview.net/forum?id=rJXMpikCZ.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation Learning on Graphs with Jumping Knowledge Networks, June 2018. URL http://arxiv.org/abs/1806.03536. arXiv:1806.03536 [cs, stat].

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks?, February 2019. URL http://arxiv.org/abs/1810.00826. arXiv:1810.00826 [cs, stat].