Beyond Graph Neural Networks Expressivity: Topological Learning for Classical Planning

Valerio Borelli[a,b,*] **and Massimiliano Tummolo**[a,b,**]

[a]University of Brescia
[b]Sapienza University of Rome

Abstract. We consider the problem of learning heuristics for classical planning domains using Graph Neural Networks (GNNs). This problem has been approached multiple times from different perspectives and with varying results for classical planning. A key limitation in this context is the expressivity of GNNs, which is restricted to the C2 fragment of first-order logic (i.e., formulas with only two variables, possibly reused). This expressivity bound implies that GNNs may not capture certain structural properties of more complex planning problems. We aim to extend the work of Ståhlberg et al. [9] to better capture the structural complexity of planning domains.

1 Introduction

In recent years, different works have shown how deep learning approaches can be used to solve planning problems, either as heuristic functions for classical planning, or applied in the context of generalized planning. For classical planning in particular, interesting results have been shown by general policies approaches [9, 10, 11], which use different types of graph neural network architectures to learn a generalized policy for a classical planning domain, and by heuristic approaches such as hypergraph networks [8] and relational heuristic networks [7]. Graph Neural Networks (GNNs) in particular demonstrated significant promise in learning heuristic functions and general policies for classical planning problems. The work of Ståhlberg et al. [10] extends GNNs for learning value functions in classical planning scenarios, showcasing their ability to exploit relational structures inherent in planning states. However, a critical limitation when using GNNs in this context is their expressivity, which has been shown to be equivalent to the C2 fragment of first-order logic, that includes formulas with at most two variables and counting quantifiers [1, 4]. This means that while GNNs can effectively capture local relational patterns and simple object interactions, they struggle to represent global structures or more complex dependencies that involve multiple objects and higher-order relations. To address this limitation, we propose incorporating topological tools, such as simplicial neural networks and persistent homology-based loss functions, that enable the modeling of n-ary interactions and global structural constraints in a more expressive way. These techniques allow us to go beyond the C2 expressivity barrier, enriching the learning of heuristics and generalized policies for classical planning. We explore these ideas in more detail in Section 4.

* Corresponding Author. Email: valerio.borelli@unibs.it.
** Email: massimiliano.tummolo@unibs.it.

2 Background

Classical Planning. A classical planning problem is a tuple $\Pi = \langle s_0, A, G, P \rangle$ where P is the set of boolean variables (or fluents), $s0$ is the initial state of the problem, A is a set of actions, and G is the goal of the problem, which consists in a set of propositional goal conditions. An action $a \in A$ is a pair $\langle Pre(a), Eff(a) \rangle$, in which $Pre(a)$ represents the set of preconditions that must be verified in a state s before executing the action, and $Eff(a)$ represents the set of effects of the action, that will be applied to the actual state to reach the next state. A plan $\pi = \langle a_1, ..., a_n \rangle$ is a sequence of actions, and is considered a solution for the planning problem if applying the sequence of actions, starting from the initial state of the problem, leads to a final state in which all the goals of the problem are satisfied.

Graph Neural Networks. A Graph Neural Network (GNN) is a type of neural network specifically designed to work with graph-structured data. Traditional neural networks are designed to represent data structured in grids or sequences, but they struggle in real-world datasets structured as graphs, like social networks or molecular structures. GNNs operate on graph-structured data by iteratively updating node embeddings through local message passing [5]. At each layer, a node aggregates messages from its neighbors and updates its representation. Despite their effectiveness, standard GNNs are limited to patterns expressible in C2 [1, 4] logic, which hampers their performance on planning domains requiring global reasoning.

GNNs for Classical Planning. A crucial design choice when applying GNNs to planning is the state encoding, that is, how to represent planning states as graphs for the network. The work by Horčik et al. [6] offers a systematic comparative analysis of different state encodings, evaluating their impact on heuristic learning for classical planning tasks. They consider three main encoding schemes:

- **Atom-based encoding:** each grounded predicate (atom) is a node; edges link atoms that share one or more objects. While this encoding preserves logical information at the level of predicates, it can result in large and sparse graphs.
- **Object-based encoding:** each object is a node, and edges are added between objects that co-occur in the same atom. Unary predicates become node labels. This encoding yields more compact graphs and performs better in terms of evaluation speed and coverage.
- **Mixed (object-atom) encoding:** both objects and atoms are nodes; edges connect each object to the atoms it appears in. This encoding captures full relational structure but can be redundant and computationally heavier.

The empirical results provided show an improvement in coverages with object-based encoding and also a reduction in the number of expanded states and evaluation time compared to atom-based or object-atom mixed encodings.

However, despite its practical advantages, the object-based encoding suffers significantly from the expressivity limitations imposed by the C2 fragment. Since each edge in the graph corresponds to a binary co-occurrence of objects within a shared predicate, higher-arity relations (e.g., ternary or quaternary atoms) are decomposed into multiple pairwise interactions. This decomposition leads to a loss of structural information: the network is no longer able to distinguish between genuinely ternary relations and sets of unrelated binary ones. As a result, GNN architectures operating over object-based encodings are inherently unable to represent or reason about complex relational structures involving more than two objects.

Ståhlberg et al. [10] proposes an approach to overcome this limitation. Specifically, they show that it is possible to extend the expressivity of GNNs beyond C2 by enriching the input state representations with derived atoms, which encode higher-order relations such as role compositions and transitive closures. These derived atoms act as additional logical features that allow GNNs to handle structures that cannot be directly represented within C2, enabling the learning of general policies even in complex domains like Logistics. However, despite their utility, derived atoms alone are not sufficient to fully capture the complete structural complexity of many planning problems. The reliance on local relational patterns and limited variable interactions means that certain global properties, such as complex dependencies involving many objects simultaneously, may still remain out of reach.

3 Topological Methods for Deep Learning

Topology provides a formal language for describing the global structural properties of data. In the context of machine learning, topological methods allow us to go beyond local, pairwise relationships (as typically modeled by GNNs) and capture higher-order interactions through constructions like simplicial complexes. A simplicial complex generalizes a graph by modeling not just vertices and edges (0- and 1-simplices), but also triangles (2-simplices), tetrahedra (3-simplices), and higher-dimensional interactions.

In particular, simplicial homology gives us algebraic invariants, such as connected components, cycles, and voids, that describe the topological structure of a space. Persistent homology extends this by analyzing how topological features evolve across scales in a filtration, yielding persistence diagrams or barcodes that summarize topological features with associated lifetimes.

Recent work, such as by Ebli et al. [2], has leveraged these ideas to define Simplicial Neural Networks (SNNs). These networks generalize GNNs to simplicial complexes and allow for message passing not only between nodes (0-simplices), but across higher-dimensional simplices via coboundary and Laplacian operators. Convolutions are defined spectrally using the eigenbasis of simplicial Laplacians, yielding localized, degree-limited filters that can propagate higher-order structural information.

Parallelly, Gabrielsson et al. [3] introduced a differentiable topology layer based on persistent homology that can be embedded within neural architectures. This layer can be used as a loss function or regularizer, allowing models to enforce topological priors or constraints, e.g., promoting connected components or holes in outputs, or encouraging topological sparsity in parameter spaces.

4 Handling higher order relations with Topology

The expressivity of standard GNNs is limited to the C2 fragment of first-order logic, meaning they can only capture patterns involving up to two variables with counting quantifiers. This limitation prevents GNNs from reasoning about many natural properties of planning problems, especially those that involve multiple objects or complex dependencies. For instance, in domains like Logistics, achieving global goal satisfaction often requires recognizing interactions among sets of objects, not just pairs.

Topology offers a promising solution to this challenge. Simplicial complexes can naturally represent n-ary relationships and encode structural features that go beyond the representational capacity of GNNs. For example, a 2-simplex can represent a mutual interaction between three objects, and the presence or absence of such higher-order simplices provides critical relational context. A specific type of neural networks, Simplicial Neural Networks(SNNs), are designed to learn over these complexes by allowing message passing across simplices of arbitrary dimensions, governed by the Hodge Laplacian.

The Laplacian encodes both upward and downward interactions, thus capturing how lower- and higher-dimensional simplices influence each other. This mechanism enables SNNs to detect cycles, voids, and other topological features that correlate with strategic substructures in planning domains.

Moreover, topological loss functions based on persistent homology, like those proposed in [3], can be used to explicitly enforce desired structural properties in the learned representations or outputs.

Example 1. *The Rovers domain requires coordinating rovers to navigate, take images, and transmit data. It features two ternary relations, which are:* $(can\_traverse\ r\ x\ y)$ *and* $(have\_image\ r\ o\ m)$*, the first encodes that a rover* r *can move between two waypoints* x *and* y*, while the second states that the rover* r *can acquire an image of objective* o *in mode* m*. In an object-based encoding, those two relations are decomposed into pairwise link, e.g.,* $(r, x), (r, y), (x, y)$*, but in doing so we lose the distinction between a true ternary relation and a collection of binary co-occurrences. In contrast, if we use simplicial complexes to represent these atoms as 2-simplices, we can preserve their higher-arity nature, allowing the network to reason directly about the joint dependency between all the objects involved.*

Key ideas for augmenting planning graphs with topology:

- Constructing simplicial complexes from sets of grounded predicates,
- Applying persistent homology to analyze global relational patterns,
- Using simplicial convolutional layers to propagate messages across higher-order relations,
- Incorporating topological regularization to bias the learning process toward structurally coherent heuristics or policies.

By leveraging these topological techniques, we can extend the representational and reasoning capabilities of neural models in classical planning, offering a principled path beyond the C2 expressivity bottleneck.

5 Conclusion

We proposed a topological extension to graph-based heuristic learning for classical planning. Simplicial complexes, SNNs, and persistent homology enable reasoning over higher-order structures, overcoming GNN expressivity limits and opening new directions for neural planning systems.

References

[1] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. Reutter, and J.-P. Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[2] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. *arXiv preprint arXiv:2010.03633*, 2020.

[3] R. B. Gabrielsson, B. J. Nelson, A. Dwaraknath, and P. Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020.

[4] M. Grohe. The logic of graph neural networks. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–17. IEEE, 2021.

[5] W. L. Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[6] R. Horčik, G. Šír, V. Šimek, and T. Pevnỳ. State encodings for gnn-based lifted planners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26525–26533, 2025.

[7] R. Karia and S. Srivastava. Learning generalized relational heuristic networks for model-agnostic planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8064–8073, 2021.

[8] W. Shen, F. Trevizan, and S. Thiébaux. Learning domain-independent planning heuristics with hypergraph networks. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 574–584, 2020.

[9] S. Ståhlberg, B. Bonet, and H. Geffner. Learning generalized policies without supervision using gnns. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 474–483, 2022.

[10] S. Ståhlberg, B. Bonet, and H. Geffner. Learning general optimal policies with graph neural networks: Expressive power, transparency, and limits. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 629–637, 2022.

[11] S. Toyer, F. Trevizan, S. Thiébaux, and L. Xie. Action schema networks: Generalised policies with deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.