Enhancing Graph Neural Network for Boolean Satisfiability Solving via Data Augmentation

Yi Fu [1] **Anthony Tompkins** [1] **Yang Song** [1] **Maurice Pagnucco** [1]

Abstract

Boolean Satisfiability (SAT) is a well-known NP-complete problem that lies at the core of many applications in formal verification, planning, and artificial intelligence. While classical SAT solvers have achieved impressive results on both synthetic and industrial benchmarks, they solve each instance independently without leveraging prior experience. Graph Neural Network (GNN)-based SAT solvers offer a learning-driven approach with the potential to transfer knowledge across problem instances and achieve better generalization performance. However, most existing work in this area focuses heavily on model architecture designs, with limited attention paid to data augmentation techniques that could improve out-of-domain generalization. In this work, we explore two simple data augmentation strategies applied during training and analyze their impact on both in-domain accuracy and out-of-domain generalization. Our findings suggest new directions for enhancing the performance and generalization ability of GNN-based SAT solvers.

1. Introduction

The Boolean Satisfiability (SAT) problem (Biere et al., 2009) is a core problem of computer science research. Theoretically, it has advanced the understanding of computation and the strategies required to tackle challenging NP hard problems. Practically, SAT has found applications in a wide range of real-world domains, including logistics planning (Kautz & Selman, 1999), software verification (Ivančić et al., 2008), and product configuration (Sinz et al., 2003).

Given the critical importance of the SAT problem, modern SAT solvers, particularly conflict-driven clause learning (CDCL) solvers (Marques-Silva et al., 2021), have been carefully designed to efficiently handle large-scale industrial instances. Despite their strong performance, the development of effective heuristics for traditional SAT solvers remains a challenging and time-consuming task, often requiring substantial domain expertise and extensive empirical tuning. Moreover, these solvers typically approach each problem independently and are tailored for specific instance types, limiting their ability to transfer knowledge across problem domains (Alyahya et al., 2023). Given these challenges, graph neural networks offer a complementary, data-driven paradigm for SAT solving. By learning patterns from previously solved instances, GNN-based approaches can potentially generalize and accelerate the solving process for new problems, which opens up new possibilities for building adaptable solvers across diverse SAT domains.

Despite growing interest and rapid progress in GNN-based SAT solving, developing effective models in this area remains a complex challenge and involves navigating a broad design space. These include choices around model architecture (e.g., GCN (Angne et al., 2021) and GAT (Chang et al., 2022a)), SAT domain selection (e.g., random, industrial, and pseudo-industrial (Alyahya et al., 2023)), task formulation (e.g., satisfiability prediction (Cameron et al., 2020), solution prediction (Selsam et al., 2019) and variable prediction (Wang et al., 2024)), and graph representation (e.g., unipartite (Mull et al., 2016), bipartite (Li et al., 2024) and tripartite (Zhang et al., 2024)). While each of these aspects poses its own challenges, most existing GNN-SAT work has concentrated on model architectural innovations only, with relatively limited attention given to dataset construction and domain selection (Zhang et al., 2022; Chang et al., 2022b; Shi et al., 2022; Cameron et al., 2020; Hartford et al., 2018). Since SAT problem domains differ significantly in their structural properties, models trained on a single domain may achieve strong in-domain performance but struggle to generalize to other domains (Li et al., 2024). This gap limits the broader applicability of GNN-based SAT solvers and stands in contrast to one of the core motivations for using GNNs in SAT solving: their potential to learn generalizable patterns from structurally diverse instances.

To address this limitation, data augmentation is both benefi-

[1]Department of Computer Science and Engineering, University of New South Wales, Sydney, Australia. Correspondence to: Yi Fu <yi.fu.1@student.unsw.edu.au>.

The second AI for MATH Workshop at the 42nd International Conference on Machine Learning, Vancouver, Canada.

cial and essential. Selecting structurally compatible training domains or incorporating more structural features can provide critical contextual information that are otherwise difficult for GNNs to learn from raw graph structure alone. These strategies help the model better capture global graph properties, ultimately improving its generalization to structurally different SAT instances. Thus, building on two recent works (Li et al., 2024; Fu et al., 2025), we investigate GNN SAT solving from the perspective of data augmentation. In particular, we explore how data preparation through the use of diverse problem domains and the inclusion of structural features can enhance both performance and generalization in GNN-based SAT solvers.

The main contributions of this paper are as follows.

- We introduce two simple yet effective data augmentation strategies for GNN-based SAT solving: (a) a novel GNN component that integrates global structural features of SAT instances, and (b) a training scheme that leverages two domains to enhance generalization.
- We conduct extensive experiments across multiple GNN models to demonstrate the effectiveness of these strategies in improving both in-domain performance and out-of-domain generalization; and
- We conclude with a discussion of future research directions, highlighting opportunities to further enhance the performance of GNN-SAT solvers.

2. Related Work

GNN SAT Solving. GNNs have been applied to SAT solving mainly as problem solvers, including standalone solvers, which are models trained to classify satisfiability directly (Selsam et al., 2019; Zhang et al., 2022; Chang et al., 2022b; Cameron et al., 2020; Hartford et al., 2018; Duan et al., 2022; Ozolins et al., 2022), and hybrid solvers, which use GNNs to guide traditional solvers by replacing specific heuristics with predictions from learned tasks. These tasks include UNSAT core prediction (Selsam & Bjørner, 2019), glue clause detection (Han, 2020) and backbone variable detection (Wang et al., 2024).

SAT structural properties. It is widely believed that different SAT domains have distinct underlying structures (Alyahya et al., 2023), since traditional solvers perform differently on random, crafted, and industrial problems. These structures include both graph-based or problem-based properties (e.g., phase transitions (Cheeseman et al., 1991), backdoors and backbones (Kilby et al., 2005), scale-free (Ansótegui et al., 2009), community structure (Ansótegui et al., 2019)) and solver-based properties such as mergeability and resolvability (Zulkoski et al., 2018). These properties have been widely used in improving traditional solvers (Audemard & Simon, 2009), improving portfolio approaches, which select the best solver to be run on a SAT problem (Xu et al., 2007), classifying benchmarks (Ansótegui et al., 2017), defining problem hardness (Newsham et al., 2014), and generating instances, especially instances similar to industrial problems (Giráldez-Cru & Levy, 2017). However, most existing work focuses on individual properties and is only evaluated using traditional SAT solvers (Li et al., 2021; Zulkoski et al., 2018) rather than GNN-based SAT solvers. Furthermore, graph based properties are often calculated on representations like VIG and CVIG (Ansótegui et al., 2019; Xu et al., 2007), which differ from the LCG graph representation commonly used in GNN-SAT solvers. There has been some exploration of structural properties in GNN-based SAT research, such as SAT solver selection (Zhang et al., 2024). However, these works do not focus on improving the performance of GNN-SAT solvers themselves.

Data Preparation in GNN-SAT. While most GNN-based SAT solvers are trained and evaluated on the same data domain without any data-level modifications, several recent works have proposed approaches that involve modifying or augmenting the input data or graph representation. (Cameron et al., 2020) encode CNF formulas into permutation-invariant sparse matrices to better capture logical structure. (Duan et al., 2022) introduce six label-preserving augmentations (LPA) on LCG instances, including techniques such as unit propagation (UP), adding unit literals (AU), and variable elimination (VE). (Fu et al., 2025) investigate the generalization ability of GNNs across 11 problem domains by analyzing model performance with respect to the structural properties of the input formulas.

3. Methodology

3.1. Preliminary

The Boolean Satisfiability (SAT) problem involves determining whether there exists an assignment of values (i.e., true or false) to a set of propositional variables V that makes a given boolean formula satisfiable. A SAT problem is usually expressed in conjunctive normal form (CNF), which is a conjunction of clauses c containing a disjunction of literals l. A literal l is either a variable $v \in V$ or its negation $\neg v$ (or $\overline{v}$). A clause c is a disjunction of literals, expressed as $(l_1 \vee l_2 \vee ..l_n)$. A formula f is a conjunction of clauses $(c_1 \wedge c_2 \wedge ...c_n)$, where each clause must be true in order for the problem to be satisfiable. For example, the formula

$$(v_1 \vee \neg v_2 \vee v_3) \wedge (v_1 \vee v_2) \wedge (v_3)$$

is satisfiable, since the assignment $v_1 = \text{True}, v_2 = \text{False}, v_3 = \text{True}$ makes the entire formula evaluate to true. Since every propositional formula can be transformed into an equivalent formula in CNF, we mainly use this form in this work to represent a SAT problem.

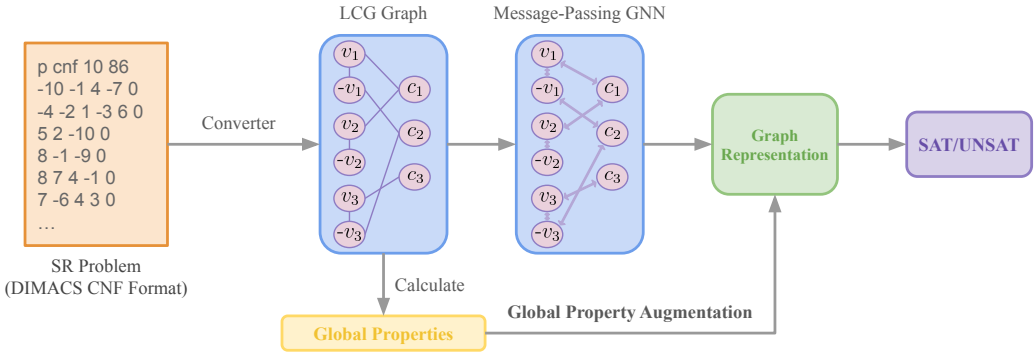

Figure 1. Overview of the global attribute component added into standard GNN message passing. Each SR problem is first represented in DIMACS CNF format, which is then converted into LCG representation. We compute global structural properties from the graph and concatenate them with the final graph embedding at the last layer of the model, alongside the output of the message passing network.

SAT formulas can be represented as graphs using various encoding schemes (Alyahya et al., 2023), including Variable Incidence Graphs (VIG), Variable Clause Graphs (VCG), Literal Incidence Graphs (LIG), and Literal Clause Graphs (LCG). In this work, we adopt the LCG representation, which is a bipartite graph with two node types: literals and clauses. An edge connects a literal node to a clause node if the literal appears in that clause. Additionally, each literal is connected to its complementary literal (e.g., v to $\neg v$). During each round of GNN message passing, the clause nodes receive messages from their connected literals, and the literal nodes receive messages from both their connected clauses and their complementary literals (Li et al., 2024).

3.2. Data Domains Selection

There are various SAT problem domains, each exhibiting distinct structural properties. To investigate our augmentation strategies effectively, we design experiments that focus on a wide range of domain types.

SR(n) (Selsam et al., 2019) is a synthetic dataset with n number of variables. It contains balanced pairs of satisfiable and unsatisfiable problems, differing by only one literal in a single clause. Each problem is generated by sampling a clause size k (with mean slightly above 4), selecting k variables uniformly at random, and negating each with 50% probability. Clauses are incrementally added and checked for satisfiability using MiniSAT (Sorensson & Een, 2005). When an unsatisfiable instance u_1 is found, flipping a literal in the final clause yields a satisfiable counterpart u_2, thus forming a closely related problem pair.

For multi-domain training and testing, we evaluate across several domain types:

- **Random domains**: Random 3-SAT (R3), where each clause contains at most 3 number of literals. This domain is generated using CNFgen (Lauria et al., 2017).
- **Combinatorial domains**: We choose 3 combinatorial problem domains in this work, which includes k-Clique (KCL), k-Dominating Set (KD), and k-Vertex Cover (KV), which are generated using CNFgen. For these problems, the goal is to find some combination of elements in a solution space while respecting defined constraints.
- **Pseudo-industrial domains**: Community Attachment (CA) (Giráldez-Crú & Levy, 2015) is a domain that mimics the community structure of industrial problems in VIG graph. Popularity-Similarity (PS) (Giráldez-Cru & Levy, 2017) measures both the community structure and the scale-free structure on a SAT instance's VCG graph.

3.3. Data Augmentation Strategies

To enhance the generalization capability of GNN-based SAT solvers, we propose two simple yet effective data augmentation strategies: incorporating global structural attributes and training with multiple domains. An overview of our approach is illustrated in Figures 1 and 2.

3.4. Global Attributes Component

Most GNN-SAT solvers convert CNF SAT formulas into graph representations and use these graphs as the sole input to the model. As a result, model performance relies entirely on the GNN's ability to learn patterns from the input graph structure. However, a single SAT formula contains many interpretable features that are not easily captured through message passing alone. This reliance on learned graph patterns can restrict a model's generalization ability.

Table 1. Descriptions of graph-level features used in our work.

Attribute	Symbol	Description
Number of clauses	n_c	Number of clauses in a formula.
Number of variables	n_v	Number of variables in a formula.
Ratio	n_c/n_v	Ratio of number of clauses to number of variables in a formula.
Self-similarity	D_f	Self-similarity is measured by the fractal dimension (D_f) (Ansótegui et al., 2014). A graph G is considered self-similar if the minimum number of boxes of size s needed to cover it decreases polynomially with D_f.
Scale-free	α_v	Measures whether the node degrees follow a power-law distribution (Ansótegui et al., 2009), which can be approximated by variable frequency (α_v).
Entropy	H	One-dimensional entropy (Zhang et al., 2021) measures uncertainty in graph structure, computed based on the graph volume vol_G.
Community structure	Q	Measured by modularity Q (Ansótegui et al., 2019), which compares within-community edges in relation to another random graph that has an equal number of vertices and degree.
Treewidth	T_w	Indicates how tree-like a graph is (Mateescu, 2011), which can be calculated using the 'treewidth-min-degree' heuristic from NetworkX (Hagberg & Conway, 2020).
Centrality	B_e	Measures node importance via betweenness centrality (Freeman, 1977), computed using NetworkX (Hagberg & Conway, 2020).

One potential solution is to augment the graph structure representation itself, for example by adding edges between nodes that share certain high-level relationships (e.g., nodes within the same community structure). While this can be effective, it requires careful data design and significant preprocessing time, and may not be applicable to all SAT problems. For instance, random SAT problems often lack clear community structure (Ansótegui et al., 2012), making such graph augmentation difficult or ineffective.

To incorporate global-level information in a simple and scalable way, we instead pre-compute a set of graph-based and CNF-based structural features and append them as a global structural input to the model. These features are scaler values, and they act as a form of high-level guidance to the model. Following prior work in structure-aware GNN SAT solving (Fu et al., 2025), we select eight structural properties that are computable across all problem domains and can be obtained within a reasonable time frame. Selected properties include number of clauses, number of variables, ratio of clause to variables, self-similarity, scale-free, entropy, community structure, treewidth and centrality. A detailed description of these attributes is provided in Table 1. We hypothesize that these structural features help the GNNs learn and generalize more effectively. Some features capture global characteristics of problem instances (e.g., number of variables), which could help generalize to all other domains, while others align with specific domains (e.g., scale-free and community structure could be particularly useful for generalizing to the PS domain).

These global properties are passed to the model alongside the original LCG graph representation. During training, we concatenate one, two, or all eight selected global features to the final layer of the model. In the case of two-feature combinations, we explore two strategies: (1) randomly selecting pairs of features (e.g., combining $D_f$and Q) and (2) selecting the well performed individual features from the single-feature experiments and combining them (e.g., n_c and n_v). The final graph-level representation is:

$$y = \sigma\left(\mathrm{MLP}\left(\left[\mathrm{MEAN}\left(\{h_l^{(T)}, l \in \mathcal{L}\right) \parallel \mathrm{Proj}(f_{\text{global}})\right]\right)\right), \tag{1}$$

where $h_l^{(T)}$ is the final hidden representation of literal node l, $\mathcal{L}$ denotes the set of all literals in the input formula, and f_{global} represents the pre-computed global structural features (e.g., number of variables). The model first applies mean pooling over all literal embeddings, then concatenates this pooled vector with the projected global features. This combined representation is passed through an MLP followed by a sigmoid activation to predict the satisfiability of the input SAT problem.

At test time, we evaluate the impact of different features and combinations of features to assess their individual and collective contributions to model performance and out-of-distribution generalization. All global feature experiments are trained on the SR domain, which has previously demonstrated strong out-of-domain generalization (Li et al., 2024; Fu et al., 2025).

3.5. Multi-Domain Training

GNN-based SAT solvers are commonly trained and evaluated on problems from a single domain. However, random, combinatorial, and industrial SAT problems differ significantly in their structural characteristics. As such, training on multiple domains, especially those spanning different domain types (e.g., random and combinatorial), has the potential to improve the model's ability to generalize beyond its training distribution. To investigate into this hypothesis, we adopt a multi-domain training strategy, which we treat

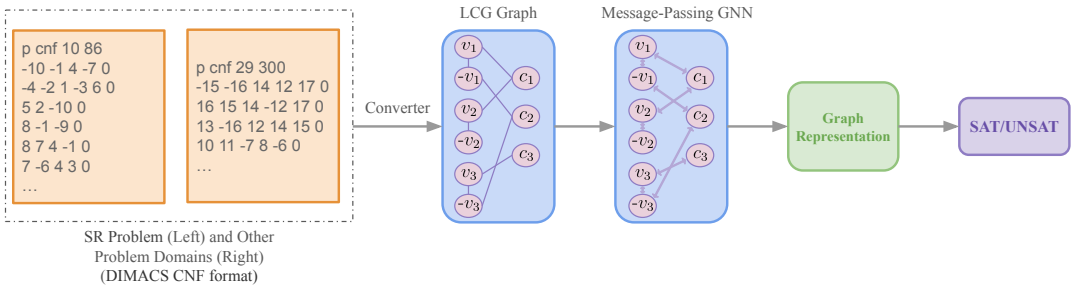

Figure 2. Overview of the multi-domain training scheme. The training dataset consists of 50% SR problems and 50% problems from one additional domain. These combined datasets are fed into the standard GNN message passing network to generate predictions for each problem instance.

as a form of domain-level data augmentation. The goal is to increase structural diversity in the training data and thereby enhance out-of-domain generalization. Specifically, we use the SR domain as a base and construct composite training datasets by mixing SR with one additional domain in equal proportion (50% SR, 50% other domain). The secondary domains include Random 3-SAT, Community Attachment, Popularity-Similarity, k-Clique, k-Dominating Set, and k-Vertex Cover.

The intuition behind this approach is as follows: since SR already demonstrates strong generalization capability, combining it with a structurally distinct domain such as KCL may help the model better capture the unique patterns of that domain class (e.g., combinatorial structures). This, in turn, could lead to improved generalization not only within that domain family but also to unrelated domains such as random or industrial SAT problems. By exposing the model to a wider range of structural patterns during training, this strategy encourages more robust representations.

4. Experiments

In this section, we present the results (reported as percentage accuracy) of applying two data augmentation strategies to GNN-based SAT solvers. Specifically, we investigate how well GNN models generalize to out-of-distribution domains when trained on the SR domain. We further examine how performance changes when introducing additional training domains, adding a single global feature, or incorporating multiple global features, compared to the baseline trained solely on SR.

4.1. Experimental Setup

We evaluate performance across several baseline GNN models: NeuroSAT (Selsam et al., 2019), GCN (Kipf & Welling, 2017) and GIN (Xu et al., 2019). All models follow the implementation of LCG graph representation provided by G4SATBench (Li et al., 2024). The learning task is satisfiability prediction, which is a graph classification problem where the model outputs a binary prediction for each input SAT instance. All models are trained on a balanced dataset consisting of 80k satisfiable and 80k unsatisfiable problems (160k in total), with an additional 10k pairs of problems used for validation. We evaluate each model on every domain separately, using 10k problem pairs per domain. This setup is chosen because each SAT domain exhibits highly distinct structural properties, and combining them during testing would obscure domain-specific performance. For example, although both pseudo-industrial domains CA and PS are crafted to mimic industrial structures, they emphasize different aspects. Thus, a model that performs well on PS may not perform equally well on CA, and vice versa, and testing on each individual domains rather than combining them is essential. All experiments are repeated three times, and we report the average accuracy across runs.

4.2. Multi-Domain Training

To evaluate whether training with multiple domains simultaneously affects performance and generalization, we train multiple GNN models using two-domain combinations. The base domain is SR(10-40), which is SR domain with 10-40 number of variables, and we construct additional training datasets by combining SR with one other domain in a 50%-50% split. Results are summarized in Table 2, where we bold the values that either outperform or are within 1% of the baseline trained on 100% SR.

As expected, the added domain usually improves test accuracy on that same domain, since it is included during training. In contrast, SR test performance tends to remain stable or slightly decrease, with a few exceptions (e.g., SR+R3 slightly improves SR performance on NeuroSAT). Interestingly, training on KCL+SR leads to a significant drop in SR accuracy for GCN and GIN. We suspect this is due to the sharp structural differences between the two domains: KCL instances are built around dense cliques with highly regular connectivity patterns while SR does not. When

trained jointly, the model may overfit to the stronger, more uniform structures from KCL, which could hinder its ability to represent SR-specific features. Additionally, since the loss is averaged over both domains, the model may optimize more aggressively for KCL if it is easier to learn, further compromising performance on SR.

Each GNN model also shows different generalization behavior. NeuroSAT performs well across most domains when trained with SR+KD or SR+CA, while SR+PS performs well for GCN and GIN. We note that training with CA+SR or PS+SR generally achieves higher accuracy across multiple test domains on all three models, suggesting that pseudo-industrial domains may help models capture more transferable structures.

Additionally, we find that feature effectiveness can also be domain-dependent. From the results, testing domains accuracies improve when trained with structurally similar problems (e.g., KCL boosts when trained with KV and vice versa). This structural similarity may be easier for GNNs to leverage during message passing.

Table 2. Overview of multi-domain training results

Train Domain	Model	Test Domains						
		R3	KCL	KD	KV	CA	PS	SR
SR	NeuroSAT	90.9	50.8	57.1	55.1	88.0	94.4	94.4
SR + KCL	NeuroSAT	**90.3**	**97.0**	50.5	**80.4**	81.2	84.9	93.2
SR + KV	NeuroSAT	**91.6**	**53.4**	53.2	**99.9**	84.2	93.2	**94.3**
SR + KD	NeuroSAT	**92.1**	**53.1**	**99.4**	**75.2**	74.3	**94.1**	**94.7**
SR + R3	NeuroSAT	**95.5**	**50.0**	50.0	50.0	**89.3**	**95.2**	**96.1**
SR + CA	NeuroSAT	**92.5**	**54.2**	**70.1**	**57.5**	**99.0**	**97.1**	93.4
SR + PS	NeuroSAT	**93.8**	**51.2**	**73.8**	**58.4**	79.5	**98.2**	95.8
SR	GCN	83.8	50.1	50.2	49.7	59.1	94.8	93.3
SR + KCL	GCN	54.7	**97.3**	**55.0**	**57.7**	46.3	61.0	53.3
SR + KV	GCN	71.6	**53.3**	**55.0**	**99.8**	57.7	84.3	83.2
SR + KD	GCN	77.2	**52.7**	**99.0**	**56.6**	**61.8**	80.0	80.2
SR + R3	GCN	**92.1**	**50.8**	47.7	45.6	51.4	**94.7**	92.1
SR + CA	GCN	66.6	**53.9**	**50.9**	48.7	**98.6**	90.9	89.4
SR + PS	GCN	**83.0**	**53.7**	41.6	44.8	**69.6**	**96.6**	91.0
SR	GIN	92.8	51.9	58.3	58.6	84.6	95.6	95.5
SR + KCL	GIN	61.5	**94.0**	51.7	**72.0**	62.4	51.7	62.1
SR + KV	GIN	87.8	**52.7**	50.8	**99.7**	53.1	87.9	90.7
SR + KD	GIN	87.2	50.4	**99.3**	**74.8**	58.4	90.2	91.1
SR + R3	GIN	**94.8**	50.4	51.2	51.7	66.1	94.4	**95.0**
SR + CA	GIN	**92.4**	49.4	**61.4**	**59.0**	**98.6**	**95.6**	94.5
SR + PS	GIN	**92.7**	49.5	**78.4**	**69.2**	69.6	**97.9**	**94.8**

4.3. Adding Single Global Feature

Table 3 shows the average testing result across different domains after concatenating one global feature to each model during training. We bold the values that outperform the baseline trained on 100% SR problems without any global structural component. As expected, no single global feature improves performance across all out-of-distribution domains, which is reasonable since each model appears to capture and rely on different aspects of graph structure.

However, some features lead to substantial improvements across many domains. For example, adding n_v to NeuroSAT, and α_v to both GCN and GIN, significantly boosts accuracy on most out-of-distribution domains. This is likely because the final graph embedding is computed via mean pooling over all literal embeddings, and both n_v and α_v are properties directly related to variable-level structure. These features likely provide high-level structural information that help the model interpret local embeddings more effectively.

The effectiveness of a feature still depends heavily on how well a model integrates it. For instance, adding n_v to GCN and α_v to NeuroSAT results in lower performance on several domains, possibly due to architectural mismatch or interference with the model's learned representations. In contrast, GIN shows highly stable behavior: most features either improve or maintain accuracy across domains, which makes GIN the most robust model under global feature augmentation.

We also observe that certain domains respond more positively to added structural features. PS and R3 consistently benefit from most global feature components in both NeuroSAT and GCN, suggesting that these domains are structurally aligned with the type of information captured by the examined global attributes. In contrast, CA exhibits high variance, particularly for GIN and GCN, where features like n_v and H result in sharp accuracy drops. This may be due to the dense community structure of CA graphs, which introduces complexity that conflicts with the provided global features, leading to overfitting or disrupted message passing.Interestingly, after appending Q, a measure of community structure, into all three models, the performance on CA drops as well. One possible reason is that the model was trained on SR, a domain without clear community structure. Thus when tested on CA, which contain strong community structure, the added modularity value could mislead the model or conflict with what it learned during training.

4.4. Adding Multiple Global Features

Table 4 shows the average accuracy obtained by adding more than one global feature during training, evaluated across different domains. Interestingly, the performance varies across the combination. Adding more features does not necessarily lead to better or worse results. For example, combining n_c and n_v consistently improves accuracy across nearly all domains and models. This aligns with earlier observations, since clauses and variables are the fundamental nodes in the graph, providing n_c and n_v as global inputs likely enriches the model's understanding of the overall structure. It is worth noting that training with all eight features concatenated to the model improves accuracy across most domains, especially on the combinatorial problems. This suggests that even without careful feature selection, the combined global attributes can still provide useful guidance to the GNN models.

In contrast, combining features like Q and D_f has little to

Table 3. Overview of results across models and testing domains with one global feature added.

Model	Feature	Testing Domains						
		R3	KCL	KD	KV	CA	PS	SR
NeuroSAT	N/A	90.9	50.8	57.1	55.1	88.0	94.4	94.4
NeuroSAT	Q	**93.8**	50.8	53.6	52.6	54.7	**96.0**	**96.2**
NeuroSAT	n_c	**93.6**	50.2	54.1	**60.9**	80.9	**94.7**	**95.9**
NeuroSAT	n_v	**94.1**	**51.1**	**57.7**	48.6	**90.6**	**97.4**	**96.4**
NeuroSAT	D_f	**94.2**	50.0	**59.0**	51.7	**91.1**	**96.1**	**96.2**
NeuroSAT	α_v	**93.1**	**51.6**	53.2	50.2	84.2	**96.2**	**96.4**
NeuroSAT	T_w	**94.2**	50.0	49.7	50.1	88.1	**96.5**	**96.4**
NeuroSAT	B_e	**93.0**	50.3	50.0	50.0	**90.8**	**96.4**	**95.7**
NeuroSAT	H	**93.3**	50.1	55.9	**57.2**	77.0	**96.3**	**96.3**
GCN	N/A	83.8	50.1	50.2	49.7	59.1	94.8	93.3
GCN	Q	**85.2**	49.5	50.1	49.7	52.2	**95.0**	92.5
GCN	n_c	82.6	49.4	50.2	50.4	54.9	94.6	92.8
GCN	n_v	83.6	50.1	49.0	47.8	**60.0**	**95.1**	92.8
GCN	D_f	**85.9**	47.4	48.7	48.7	55.7	**94.9**	92.9
GCN	α_v	**85.1**	**50.6**	**52.0**	50.3	**60.8**	**95.2**	**96.4**
GCN	T_w	**84.3**	49.6	**51.4**	**51.6**	**60.0**	94.7	93.1
GCN	B_e	**86.2**	49.0	52.8	49.7	58.8	94.8	**95.7**
GCN	H	83.8	49.4	49.5	47.4	58.3	94.8	93.0
GIN	N/A	92.8	51.9	58.3	58.6	84.6	95.6	95.5
GIN	Q	92.8	50.1	**59.7**	58.2	64.8	94.1	**95.7**
GIN	n_c	**93.0**	50.1	**61.8**	**61.9**	65.1	95.1	**95.6**
GIN	n_v	**92.9**	50.8	58.0	62.1	53.1	93.8	95.5
GIN	α_v	**93.0**	52.6	**58.8**	**60.1**	70.4	94.0	**95.9**
GIN	T_w	92.5	52.5	56.4	56.7	61.6	92.7	**96.0**
GIN	B_e	**93.2**	50.1	53.2	57.6	73.7	**96.2**	**96.0**
GIN	H	92.8	49.9	**60.2**	53.5	53.8	94.5	95.6

no impact on performance, likely because neither feature individually contributed much to generalization, as shown in Table 3. Moreover, these features do not offer sufficiently diverse or informative signals for the GNNs to learn from. This highlights that the effectiveness of multi-feature augmentation depends not only on the individual usefulness of each feature but also on how well they complement one another. As a result, careful selection of feature combinations is essential for this strategy to be effective.

Table 4. Overview of results across models and testing domains with two or all global features added.

Model	Feature	Testing Domains						
		R3	KCL	KD	KV	CA	PS	SR
NeuroSAT	N/A	90.9	50.8	57.1	55.1	88.0	94.4	94.4
NeuroSAT	$n_c + n_v$	**92.8**	50.8	**58.3**	**58.2**	85.6	**95.8**	**96.0**
NeuroSAT	all features	**92.8**	48.8	**62.8**	**69.8**	86.4	**95.0**	**95.1**
GCN	N/A	83.8	50.1	50.2	49.7	59.1	94.8	93.3
GCN	$Q + D_f$	80.6	50.1	49.2	**51.8**	51.5	94.8	92.3
GCN	$n_c + n_v$	83.8	49.5	**50.5**	**53.7**	**61.2**	**95.3**	93.2
GCN	all features	82.3	48.1	**51.7**	**52.8**	50.0	**94.9**	93.0
GIN	N/A	92.8	51.9	58.3	58.6	84.6	95.6	95.5
GIN	$Q + D_f$	**93.5**	51.6	58.0	56.8	72.3	94.2	95.9
GIN	$n_c + n_v$	**93.3**	**52.6**	55.6	**62.8**	68.8	95.1	**95.7**
GIN	all features	92.8	**54.0**	**59.4**	**65.7**	52.7	94.0	**95.7**

5. Discussion and Future Direction

Based on the results presented in Section 4, this section summarizes our key findings and outlines future research directions.

Out-of-distribution generalization remains a core challenge for GNN-based SAT solvers, largely due to the diverse structural properties exhibited by different SAT domains, whether represented as CNF formulas or different graphs. Our results, in line with prior studies, confirm that no single model or training domain consistently generalizes well across all other domains. For scenarios requiring maximum accuracy, training and testing within the same domain is still the most reliable approach. However, we find that carefully designed data augmentation strategies, thoughtful domain selection, and the use of suitable or more generalizable model architectures such as GIN can significantly improve generalization.

Multi-domain training, especially with two domains, can be beneficial when the added domain shares structural similarities with the target domain. For instance, pairing SR with a combinatorial domain such as k-clique can improve performance on related combinatorial tasks such as k-vercov. Our results also suggest that including structurally rich domains like CA or PS alongside SR tends to improve generalization to more out-of-domain problems, which is likely due to the expressive structural patterns of the pseudo-industry problems.

Additionally, incorporating global features into the training process presents another effective strategy for improving generalization. When features are carefully selected, they can offer valuable structural information to guide learning, particularly those aligned with the GNN architecture designs. For instance, we find that n_v and $\alpha_v$consistently boost performance across multiple domains and models. Moreover, combining multiple global features further enhances generalization in many domains and models, suggesting that feature complementarity can play an important role when designing data augmentation for GNN-SAT solvers. Note that in our experiments, we adopted a fixed 50%–50% ratio for multi-domain training. However, exploring alternative training ratios may further improve performance and is left for future work. Moreover, jointly applying global structural features and multi-domain training presents another promising direction for future exploration.

Although current GNN-based SAT solvers do not yet surpass traditional CDCL solvers in terms of accuracy, they show promise in handling smaller SAT instances more efficiently. Unlike traditional solvers, which are often bottlenecked by problem complexity, GNNs offer a fast, structure-aware alternative. Future work should continue to expand the diversity of problem domains used during training, explore more targeted forms of data augmentation, and refine GNN architectures tailored to different SAT domains. These steps will not only advance solver capabilities but also deepen our understanding of structural learning in the context of SAT solving.

6. Conclusion

In this work, we presented and analyzed two simple data augmentation strategies for GNN-based SAT solvers: training with mixed problem domains and adding pre-computed global structural features into the model. Our results show that, when chosen carefully, both strategies can improve out-of-distribution generalization across a range of SAT domains. However, not all combinations of domains and structures are effective, highlighting the importance of thoughtful domain selection and feature design. We believe these findings offer useful insights for future work, particularly in scenarios where training data is limited to obtain, such as industrial SAT problems.

References

Alyahya, T. N., Menai, M. E. B., and Mathkour, H. On the structure of the boolean satisfiability problem: A survey. *ACM Computing Surveys*, 2023.

Angne, H., Atkari, A., Dhargalkar, N., and Kale, D. Unravelling sat: Discussion on the suitability and implementation of graph convolutional networks for solving sat. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 2*, pp. 251–258. Springer, 2021.

Ansótegui, C., Bonet, M. L., and Levy, J. On the structure of industrial SAT instances. In *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming*, volume 5732 of *Lecture Notes in Computer Science*, pp. 127–141. Springer, 2009.

Ansótegui, C., Giráldez-Cru, J., and Levy, J. The community structure of SAT formulas. In *Theory and Applications of Satisfiability Testing – SAT 2012*, volume 7317 of *Lecture Notes*, pp. 410–423, 2012.

Ansótegui, C., Bonet, M. L., Giráldez-Cru, J., and Levy, J. The fractal dimension of SAT formulas. In *7th International Joint Conference on Automated Reasoning (IJCAR-2014)*, volume 8562, pp. 107–121. Springer, 2014.

Ansótegui, C., Bonet, M. L., Giráldez-Cru, J., and Levy, J. Structure features for SAT instances classification. *Journal of Applied Logic*, 23:27–39, September 2017.

Ansótegui, C., Bonet, M. L., Giráldez-Cru, J., Levy, J., and Simon, L. Community structure in industrial SAT instances. *Journal of Artificial Intelligence Research*, 66: 443–472, 2019.

Audemard, G. and Simon, L. Predicting learnt clauses quality in modern SAT solvers. In *Proceedings of the Twenty-first international joint conference on artificial intelligence*. Citeseer, 2009.

Biere, A., Heule, M., and van Maaren, H. *Handbook of satisfiability*, volume 185. IOS press, 2009.

Cameron, C., Chen, R., Hartford, J., and Leyton-Brown, K. Predicting propositional satisfiability via end-to-end learning. In *the Proceeding of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 3324–3331. AAAI Press, 2020.

Chang, W., Zhang, H., and Luo, J. Predicting propositional satisfiability based on graph attention networks. *International Journal of Computational Intelligence Systems*, 15 (1):84, 2022a.

Chang, W., Zhang, H., and Luo, J. Predicting propositional satisfiability based on graph attention networks. *International Journal of Computational Intelligence Systems*, 15 (1):84, 2022b.

Cheeseman, P. C., Kanefsky, B., Taylor, W. M., et al. Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'91, pp. 331–337, 1991.

Duan, H., Vaezipoor, P., Paulus, M. B., Ruan, Y., and Maddison, C. Augment with care: Contrastive learning for combinatorial problems. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022,*, volume 162, pp. 5627–5642. PMLR, 2022.

Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry*, pp. 35–41, 1977.

Fu, Y., Tompkins, A., Song, Y., and Pagnucco, M. Structure based SAT dataset for analysing GNN generalisation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2025.

Giráldez-Crú, J. and Levy, J. A modularity-based random SAT instances generator. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 1952–1958. AAAI Press, 2015.

Giráldez-Cru, J. and Levy, J. Locality in random SAT instances. In *Proceeding of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 638–644, 2017.

Hagberg, A. and Conway, D. Networkx: Network analysis with Python. *URL: https://networkx. github. io*, 2020.

Han, J. M. Enhancing SAT solvers with glue variable predictions, 2020.

Hartford, J., Graham, D., Leyton-Brown, K., and Ravanbakhsh, S. Deep models of interactions across sets. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pp. 1914–1923. PMLR, 2018.

Ivančić, F., Yang, Z., Ganai, M. K., Gupta, A., and Ashar, P. Efficient SAT-based bounded model checking for software verification. *Theoretical Computer Science*, pp. 256–274, 2008.

Kautz, H. and Selman, B. Unifying SAT-based and Graph-based Planning. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99,*, pp. 318–325, 1999.

Kilby, P., Slaney, J., Thiébaux, S., and Walsh, T. Backbones and backdoors in satisfiability. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, pp. 1368–1373, 2005.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR*, 2017. I can't find the pages.

Lauria, M., Elffers, J., Nordström, J., and Vinyals, M. CNFgen: A generator of crafted benchmarks. In *Theory and Applications of Satisfiability Testing – SAT 2017*. Springer International Publishing, 2017.

Li, C., Chung, J., Mukherjee, S., Vinyals, M., Fleming, N., Kolokolova, A., Mu, A., and Ganesh, V. On the hierarchical community structure of practical Boolean formulas. In *Proceedings of the Theory and Applications of Satisfiability Testing–SAT 2021: 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings 24*, pp. 359–376. Springer, 2021.

Li, Z., Guo, J., and Si, X. G4SATBench: Benchmarking and advancing SAT solving with graph neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=7VB5db72lr.

Marques-Silva, J., Lynce, I., and Malik, S. Chapter 4. Conflict-Driven Clause Learning SAT Solvers. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021.

Mateescu, R. Treewidth in industrial SAT benchmarks. Technical report, Technical Report MSR-TR-2011-22, Microsoft Research, 2011.

Mull, N., Fremont, D. J., and Seshia, S. A. On the hardness of SAT with community structure. In *International Conference on Theory and Applications of Satisfiability Testing*, volume 9710, pp. 141–159, 2016.

Newsham, Z., Ganesh, V., Fischmeister, S., Audemard, G., and Simon, L. Impact of community structure on SAT solver performance. In *Proceedings of the Theory and Applications of Satisfiability Testing–SAT 2014: Proceedings of 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings 17*, pp. 252–268. Springer, 2014. I don't know how to deal with this booktitle, looks kinda strange.

Ozolins, E., Freivalds, K., Draguns, A., Gaile, E., Zakovskis, R., and Kozlovics, S. Goal-aware neural SAT solver. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2022*, pp. 1–8, 2022.

Selsam, D. and Bjørner, N. Guiding high-performance SAT solvers with Unsat-Core predictions. In *Theory and Applications of Satisfiability Testing – SAT 2019*, volume 11628 of *Lecture Notes in Computer Science*, pp. 336–353. Springer International Publishing, 2019.

Selsam, D., Lamm, M., Bünz, B., Liang, P., de Moura, L., and Dill, D. L. Learning a SAT solver from single-bit supervision. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Shi, Z., Li, M., Khan, S., Zhen, H.-L., Yuan, M., and Xu, Q. SATformer: Transformers for SAT solving. *CoRR*, abs/2209.00953, 2022.

Sinz, C., Kaiser, A., and Küchlin, W. Formal methods for the validation of automotive product configuration data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, pp. 75–97, 2003.

Sorensson, N. and Een, N. Minisat v1. 13-a SAT solver with conflict-clause minimization. *SAT*, 2005(53):1–2, 2005.

Wang, W., Hu, Y., Tiwari, M., Khurshid, S., McMillan, K., and Miikkulainen, R. NeuroBack: Improving CDCL SAT solving using Graph Neural Networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024. can't find the pages.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are Graph Neural Networks? In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, 2019.

Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. Satzilla-07: The design and analysis of an algorithm portfolio for sat. In *Principles and Practice of Constraint Programming–CP 2007: 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007. Proceedings 13*, pp. 712–727. Springer, 2007.

Zhang, C., Zhang, Y., Mao, J., Chen, W., Yue, L., Bai, G., and Xu, M. Towards better generalization for neural network-based SAT solvers. In *Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, volume 13281, pp. 199–210, 2022.

Zhang, Z., Xu, D., and Zhou, J. A structural entropy measurement principle of propositional formulas in conjunctive normal form. *Entropy*, 23(3):303, 2021.

Zhang, Z., Chételat, D., Cotnareanu, J., Ghose, A., Xiao, W., Zhen, H.-L., Zhang, Y., Hao, J., Coates, M., and Yuan, M. Grass: Combining graph neural networks with expert knowledge for sat solver selection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6301–6311, 2024.

Zulkoski, E., Martins, R., Wintersteiger, C. M., Liang, J. H., Czarnecki, K., and Ganesh, V. The effect of structural measures and merges on SAT solver performance. In *Proceedings of the Principles and Practice of Constraint Programming: 24th International Conference, CP 2018, Lille, France, August 27-31, 2018, Proceedings 24*, pp. 436–452. Springer, 2018.