Learning Unified Graph and Language Representations for SMT Algorithm Selection

Extended Abstract of the Full Paper to Appear at CP 2026

Zhengyang Lu
University of Waterloo, Canada

Paul Sarnighausen-Cahn
University of Göttingen and CIDAS, Germany

Jiahao Chen
Georgia Institute of Technology, USA

Arie Gurfinkel
University of Waterloo, Canada

Florin Manea
University of Göttingen and CIDAS, Germany

Vijay Ganesh
Georgia Institute of Technology, USA

1 Introduction

Satisfiability Modulo Theories (SMT) solvers are core engines for applications in software engineering, formal verification, and planning [2]. Since no single solver performs best across all SMT instances, different solvers exhibit complementary strengths on different problem families. This motivates machine-learning-based algorithm selection, which aims to automatically exploit solver complementarity. SATzilla [8] pioneered this idea for SAT, and MachSMT [6] extended it to SMT solving.

The effectiveness of algorithm selection depends heavily on instance representation. Early approaches, such as MachSMT, relied on handcrafted syntactic features. More recently, graph neural networks (GNNs) have emerged as a natural representation framework for SMT formulas due to their abstract syntax tree (AST) structure, as explored by Hůla et al. [3] and Sibyl [4].

However, existing approaches largely focus on syntactic representations and overlook high-level semantic context, such as the application domain or benchmark origin, which practitioners often use when selecting solvers. Motivated by this observation, we present SMT-Select, a multimodal framework that combines graph representations of formula ASTs with transformer-based embeddings of natural-language descriptions to guide solver selection, achieving state-of-the-art performance across nine SMT logics.

2 SMT-Select: Multimodal Learning-Based Algorithm Selection

SMT-Select is a multimodal SMT algorithm selector that integrates syntactic and semantic representations. Given an SMT formula and its natural-language description, the framework encodes and fuses both inputs into a joint embedding. The formula AST is encoded using a GNN, while the textual description is encoded using a pretrained all-mpnet-base-v2 model [7, 5]. The fused representation is then processed by a pairwise selector. Figure 1 provides an overview of the framework.

Training uses a two-stage cost-sensitive pairwise objective. We first train the graph encoder using only structural information, then train the multimodal fusion module while

32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).
Editor: Nicolas Beldiceanu; Article No. 9; pp. 9:1–9:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

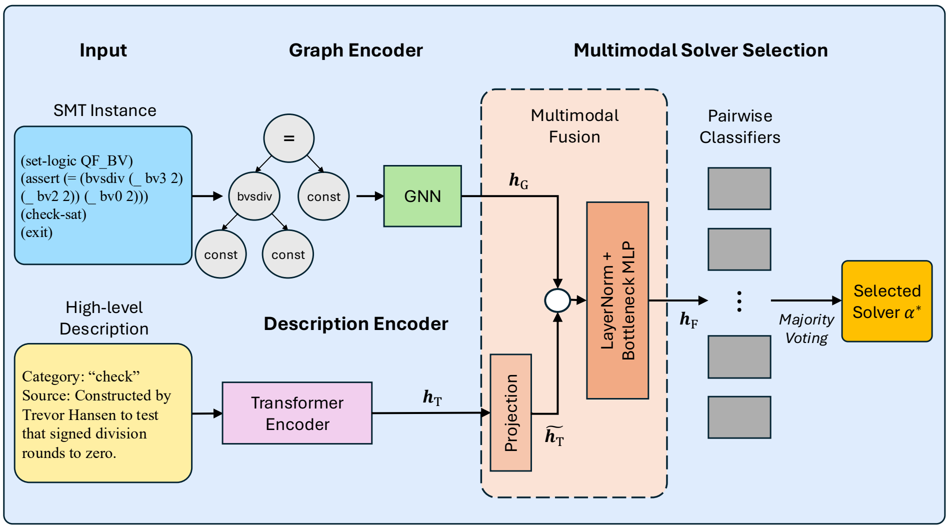

Figure 1 Overview of the SMT-SELECT architecture.

	MachSMT	Sibyl	SMT-Select (w/o desc.)	SMT-Select (w/ desc.)
ABV	7.83	17.97	76.56	**79.44**
ALIA	27.73	33.34	55.07	**67.65**
BV	17.11	12.00	42.43	**53.82**
QF_BV	7.35	-29.45	47.68	**57.52**
QF_IDL	6.06	-30.47	69.66	**81.82**
QF_LIA	84.70	76.66	**92.56**	91.65
QF_NRA	17.25	-3.50	50.61	**55.17**
QF_SLIA	76.96	—	92.82	**93.50**
UFNIA	-129.22	—	24.48	**31.84**

Table 1 PAR-2 SBS–VBS gap closed (%) on held-out test sets, averaged over five random splits. Sibyl is omitted for QF_SLIA and UFNIA due to parser limitations.

keeping the graph encoder fixed. Pairwise samples are weighted by PAR-2 runtime differences, emphasizing decisions with larger performance impact.

3 Experiment

We evaluate SMT-SELECT on nine logics from the SMT-COMP 2024 [1]. For each logic, benchmarks are split into 75% training and 25% testing sets, and natural-language descriptions are extracted primarily from SMT-LIB instance headers. To reduce variability from train–test partitioning, all results are averaged over five independent random splits.

Table 1 summarizes the PAR-2 results of SMT-SELECT and the baseline SMT selectors MACHSMT and SIBYL, reported as the proportion of the gap closed between the single best solver (SBS) and the virtual best solver (VBS). Higher values indicate better performance.

Experimentally, even the graph-only variant of SMT-SELECT consistently outperforms existing SMT selectors and SMT-COMP winning solvers. Incorporating natural-language descriptions further improves performance in eight of the nine evaluated logics.

References

1 Martin Bromberger, François Bobot, and Martin Jonáš. SMT-COMP 2024. https://smt-comp.github.io/2024/, 2024. Accessed: 2025-12-22.

2 Leonardo De Moura and Nikolaj Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011. doi:10.1145/1995376.1995394.

3 Jan Hůla, David Mojžíšek, and Mikoláš Janota. Graph neural networks for scheduling of SMT solvers. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 447–451, 2021. doi:10.1109/ICTAI52525.2021.00072.

4 Will Leeson, Matthew B Dwyer, and Antonio Filieri. Sibyl: Improving software engineering tools with SMT selection. In *Proceedings of the 45th International Conference on Software Engineering*, page 2185–2197. IEEE Press, 2023. doi:10.1109/ICSE48619.2023.00184.

5 Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL: https://arxiv.org/abs/1908.10084, doi:10.48550/arXiv.1908.10084.

6 Joseph Scott, Aina Niemetz, Mathias Preiner, Saeed Nejati, and Vijay Ganesh. MachSMT: A machine learning-based algorithm selector for SMT solvers. In *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021*, volume 12652 of *Lecture Notes in Computer Science*, pages 303–325, 2021. doi:10.1007/978-3-030-72013-1_16.

7 Sentence Transformers. all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 2021.

8 Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res.*, 32:565–606, 2008. doi:10.1613/jair.2490.