THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

NeuroSelect: Learning to Select Clauses in SAT Solvers

Hongduo Liu[1], Peng Xu[1], Yuan Pu[1], Lihao Yin[2], Hui-Ling Zhen[2], Mingxuan Yuan[2], Tsung-Yi Ho[1], Bei Yu[1]

[1]The Chinese University of Hong Kong
[2]Huawei Noah's Ark Lab

HUAWEI

Outline

Boolean Satisfiability (SAT)

- The Boolean satisfiability (SAT) problem involves finding a satisfying assignment for a Boolean formula or proving that none exists.
- SAT has wide applications in circuit verification, test pattern generation, automatic theorem proving, etc.
- SAT is the first problem proven NP-complete.

Learning for SAT

- End-to-end solvers like NeuroSAT[1]: can only handle toy cases, lack of completeness.
- Learning-aided SAT solvers: use machine learning to improve a SAT solver's heuristics like variable branching[2] and restart policy[3].

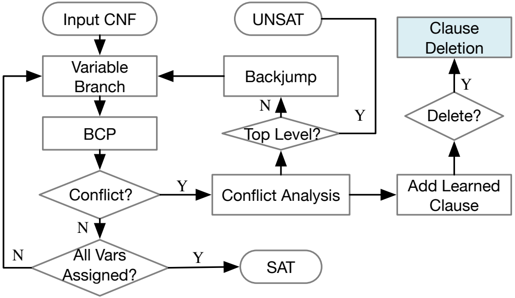
[1] Daniel Selsam, Matthew Lamm, et al. (2018). "Learning a SAT Solver from Single-Bit Supervision". In: *Proc. ICLR*.

[2] Daniel Selsam and Nikolaj Bjørner (2019). "Guiding high-performance SAT solvers with unsat-core predictions". In: *Proc. SAT*, pp. 336–353.

[3] Jia Hui Liang et al. (2018). "Machine learning-based restart policy for CDCL SAT solvers". In: *Proc. SAT*, pp. 94–110.

Clause Deletion

Clause deletion in CDCL solvers removes less useful learned clauses to manage memory and computational resources.



The flow of conflict-driven clause learning (CDCL) algorithm.

Clause Deletion Metrics in SAT Solvers

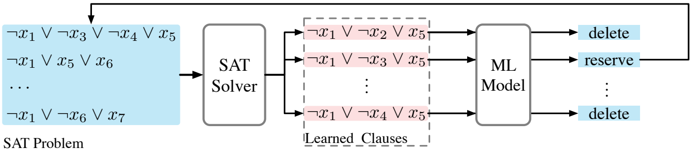
- **Activity:** Measures frequency of involvement in conflict analysis.
- **Size:** Counts the number of literals in the clause.
- **Glue Value:** Indicates diversity of decision levels involved in the clause.

Human-designed heuristics are utilized to guide clause deletion. Is it possible to develop a more effective heuristic through learning?

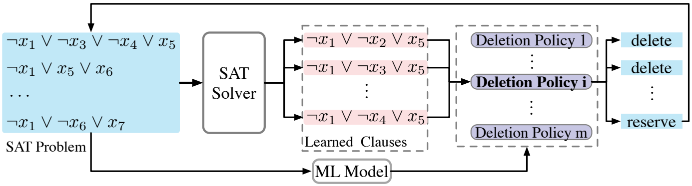
Hardness of Clause Evaluation Using ML

- **Dynamic Solver State:** A SAT solver's state changes frequently as it navigates to a new search space.
- **Inter-clause Dependencies:** The value of a learned clause depends on its interaction with other chosen learned clauses.
- **Clause Evaluation Cost:** Direct clause evaluation demands model inferences for each learned clause.

From Clause Evaluation to Policy Evaluation



(a)



(b)

Two learning aided clause deletion mechanisms. (a) Evaluate learned clauses directly; (b) Evaluate clause deletion policies.

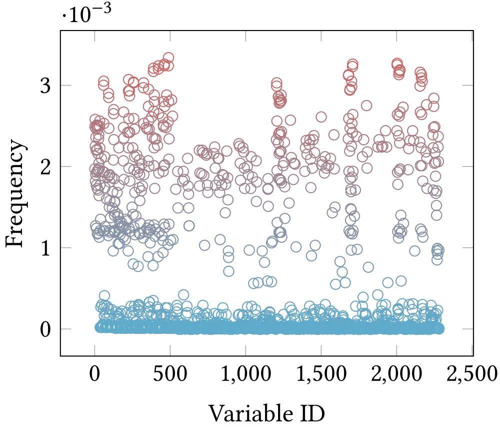
Clause Deletion Policy Evaluation

- Effectiveness of clause deletion depends on both characteristics of each learned clause and the deletion policy.
- Clause deletion policy has a lifelong effect during SAT solving.
- Evaluating the deletion policy only requires one-time inference, it can be efficient even on CPUs.

Contributions

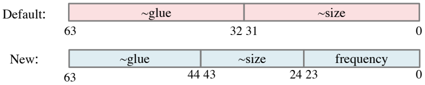
- **Step 1: Generate a Complementary Clause Deletion Policy**
  - Introduced a novel clause deletion metric based on the frequency of variable propagation.
- **Step 2: Select the Most Suitable Clause Deletion Policy**
  - Developed a classification network utilizing local message passing and global attention mechanisms.

A New Clause Deletion Metric



Distribution of variable propagation frequency of a SAT instance from SAT competition 2022. Some variables are propagated significantly more frequently than others.

A New Clause Deletion Policy

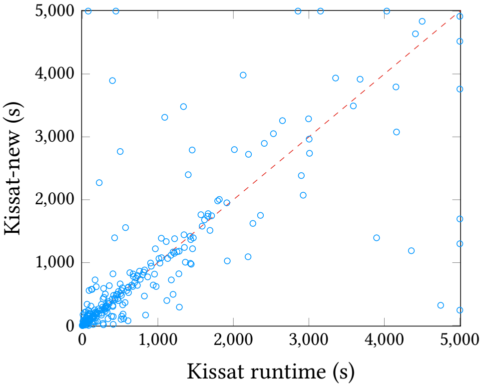


The default learned clause scoring algorithm in Kissat vs. Our new learned clause scoring algorithm considers variable propagation frequency.

$$\text{c.frequency} = \sum_{v \in c} (f_v > \alpha f_{\max}).$$

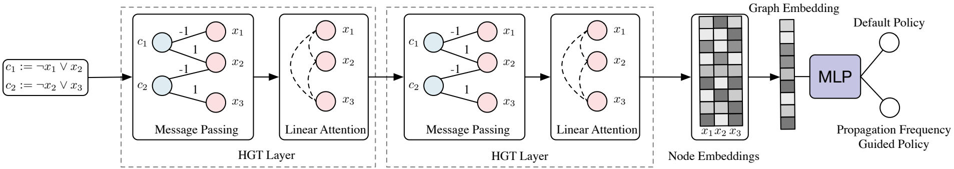
- f_v indicates the frequency of variable v used to trigger propagation since the last clause deletion.
- $f_{\max}$ represents the maximum propagation frequency among all variables.
- α is a hyperparameter between 0 and 1.

Performance of New Clause Deletion Policy



Runtime Comparison between the default and new clause deletion policy on SAT competition 2022 instances using a standard 5,000 seconds timeout.

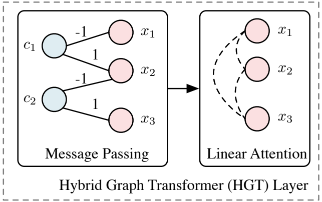
Overview



Overview of NeuroSelect.

- Every SAT instance is represented as a weighted bipartite graph.
- The weight is -1 when the variable is negated in the clause.

Hybrid Graph Transformer



Every HGT layer consists of a message-passing layer and a linear attention layer.

- The message-passing comprehends the structural information of the CNF formula.
- The linear attention captures long-term dependencies between variables.
- Linear attention reduces traditional self-attention complexity from quadratic to linear.

Linear Attention

Assume the input node embedding of the linear attention layer is denoted by $\boldsymbol{Z} \in \mathbb{R}^{N\times d}$. The linear attention function[4] is defined as

$$\begin{aligned}
\boldsymbol{Q} &= f_Q(\boldsymbol{Z}), & \tilde{\boldsymbol{Q}} &= \frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_F}, & \boldsymbol{V} &= f_V(\boldsymbol{Z}), \\
\boldsymbol{K} &= f_K(\boldsymbol{Z}), & \tilde{\boldsymbol{K}} &= \frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_F}, & \boldsymbol{D} &= \operatorname{diag}\left(\mathbf{1} + \frac{1}{N}\tilde{\boldsymbol{Q}}\left(\tilde{\boldsymbol{K}}^\top \mathbf{1}\right)\right),
\end{aligned} \tag{1}$$

where f_Q, f_K, and f_V are linear feed-forward layers to encode the query, key, and value matrix. Subsequently, we have the output of the global attention layer in the format of

$$\boldsymbol{Z}^{out} = \operatorname{LinearAttn}(\boldsymbol{Z}) = \boldsymbol{D}^{-1}\left[\boldsymbol{V} + \frac{1}{N}\tilde{\boldsymbol{Q}}\left(\tilde{\boldsymbol{K}}^\top \boldsymbol{V}\right)\right]. \tag{2}$$

[4] Qitian Wu et al. (2023). “SGFormer: Simplifying and Empowering Transformers for Large-Graph Representations”. In: *Proc. NIPS*.

Linear Attention

$$\boldsymbol{Z}^{out} = \mathrm{LinearAttn}\left(\boldsymbol{Z}\right) = \boldsymbol{D}^{-1}\left[\boldsymbol{V} + \frac{1}{N}\tilde{\boldsymbol{Q}}\left(\tilde{\boldsymbol{K}}^{\top}\boldsymbol{V}\right)\right]. \tag{3}$$

- The dimension of $\tilde{\boldsymbol{K}}^{\top}$ is $d \times N$ and the dimension of $\boldsymbol{V}$ is $N \times d$.
- The complexity of $\tilde{\boldsymbol{K}}^{\top}\boldsymbol{V}$ is $N \times d^2$.
- Give $d^2 \ll N$, the complexity is linear to N.

Datasets

Table: Statistics of the training and test datasets from SAT competitions.

Data Type	Year	# CNFs	# Variables	# Clauses
Training	2016	74	16,649	86,186
	2017	124	12,863	99,896
	2018	148	13,407	93,094
	2019	131	12,237	68,900
	2020	123	16,921	85,808
	2021	136	16,219	97,434
Test	2022	144	19,807	104,472

- An SAT instance is labeled as '1' if it sees at least a 2% reduction in propagations with the new deletion policy compared to the default policy in Kissat; otherwise, it is labeled as '0'.
- Any formula whose graph conversion exceeds 400,000 nodes is excluded to adhere to GPU memory limitations.

Classification Capability of NeuroSelect

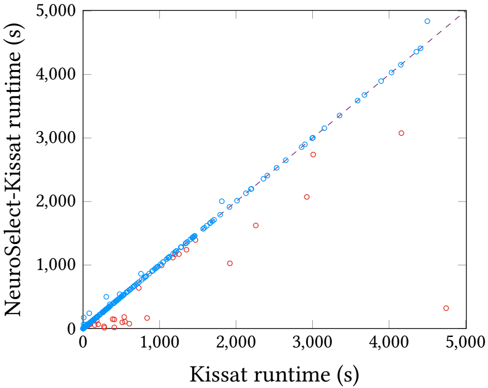
Table: Performance of different SAT classification models.

	precison	recall	F1	accuracy
NeuroSAT[5]	47.27%	44.07%	45.61%	56.94%
G4SATBench[6]	43.48%	33.90%	38.10%	54.86%
NeuroSelect w/o attention	56.45%	**58.33%**	57.38%	63.89%
NeuroSelect	**66.00%**	55.00%	**60.50%**	**69.44%**

[5] Daniel Selsam, Matthew Lamm, et al. (2018). “Learning a SAT Solver from Single-Bit Supervision”. In: *Proc. ICLR*.

[6] Zhaoyu Li, Jinpei Guo, and Xujie Si (2023). “G4SATBench: Benchmarking and Advancing SAT Solving with Graph Neural Networks”. In: *arXiv preprint arXiv:2309.16941*.

NeuroSelect-Kissat



Comparisons between NeuroSelect-Kissat and Kissat on SAT competition 2022 instances.

NeuroSelect-Kissat

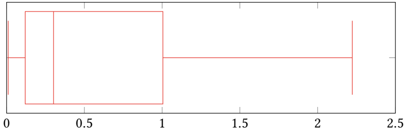
Table: Runtime statistics of Kissat and NeuroSelect-Kissat on SAT competition 2022 instances.

	solved	median (s)	average (s)
Kissat[7]	274	307.02	713.28
NeuroSelect-Kissat	**274**	**271.34**	**671.73**

Both NeuroSelect-Kissat and Kissat solved 274 instances within the time limit. However, NeuroSelet-Kissat has a shorter average runtime, leading to a 5.8% improvement.

[7] Armin Biere and Mathias Fleury (2022). "Gimsatul, IsaSAT and Kissat entering the SAT Competition 2022". In: *Proc. of SAT Competition*.

Runtime Analysis



Box plot of model inference time (s)

Inference time varies between 0.01 and 2.22 seconds on the CPU, which can be ignored compared with SAT solving time.

THE CHIPS TO SYSTEMS CONFERENCE

SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

HUAWEI

Thanks!