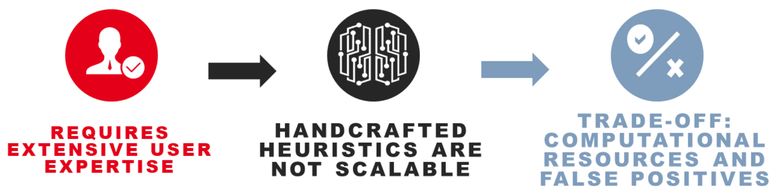
Loupe: End-to-End Learning of Loop Unrolling Heuristics for Abstract Interpretation

Maykel Mattar[1,2], Michele Alberti[1], Valentin Perrelle[1], Salah Sadou[2]

1. *Université Paris-Saclay, CEA, List, Palaiseau, France*, 2. *Université Bretagne Sud, IRISA, Vannes, France*

Context

- **Abstract Interpretation** is crucial for software verification and bug detection, but false alarms limit practical adoption
- State-of-the-art analyzers (e.g. Frama-C/Eva, Mopsa, Astrée, etc…) employ **precision-enhancing** strategies such as loop unrolling
- These strategies are computationally **expensive** and require **manual parameterization**, which is challenging even for domain experts
- Prior approaches:
  - **Per-program** configuration via iterative tuning (time-consuming)
  - **ML-based** heuristics relying on handcrafted features (poor generalization)
- **Research Gap**: Need for **automated**, **general, feature-free** strategy parameterization



Motivating Example

```
1.  #include <stdio.h>

2.  int fib(int n) {
3.      int a = 1, b = 1;
4.      //@ loop unroll 100;
5.      for (int i = 3; i <= n; i++) {
6.              int tmp = a;
7.              a += b;
8.              b = tmp;
9.      }
10.     return a;
11. }

12. void main() {
13.     //@ loop unroll 0;
14.     for (int i = 1, n; i <= 10; i++) {
15.         printf("Enter a number <= 30: ");
16.         scanf("%d", &n);
17.         if (n > 0 && n <= 30) {
18.             printf("fib(%d)=%d\n", n, fib(n));
19.             break;
20.         }
21.     }
22. }
```

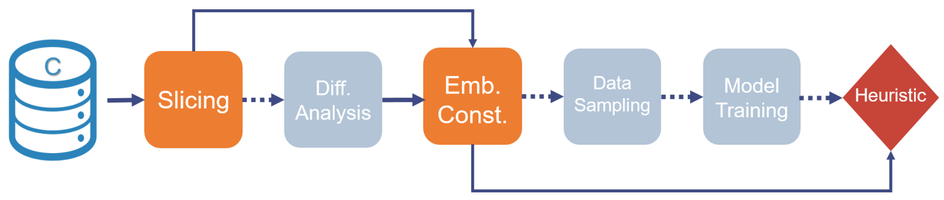
Frama-C/Eva is not requested to unroll the loop:

- ***a*** and ***b*** are in ***[1, 2^{31} − 1]***
- Overflow alarm upon ***a += b***

Frama-C/Eva is requested to unroll the loop:

- **Per-iteration** analysis of loop at line 4
- Precise approximation of **a** and **b** ranges
- The **false alarm disappears**

The Loupe Framework



- Large-scale, **self-contained** C files
- **Extracts** loops with dependencies; focuses on loop behavior
- Uses analyzer as **oracle**; labels loops by **comparing** with/without unrolling
- **CPG** for graph models or **IR2Vec/Sparrow** for vector approaches
- **Benchmarks** XGBoost, SOTA, CodeLLMs, and GNNs

Efficacy

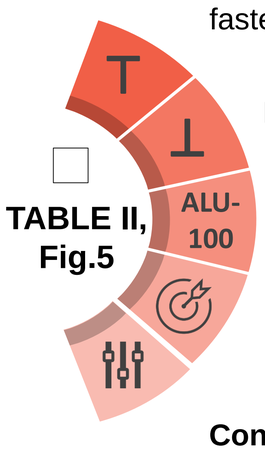
RQ1:

☐ **Loop unrolling heuristic learning comparison on preprocessed and real-world datasets. TABLE I**

Model	ANGHABENCH (Test Split)				OSCS			
	P / R	F1	F2	B-ACC	P / R	F1	F2	B-ACC
Random	8.4 / 50.3	14.3	25.1	50.1	11.5 / 48.2	18.6	29.5	49.2
XGBOOST- IR2VEC	28.3 / 33.5	30.7	32.3	62.7	54.9 / 14.7	23.2	17.2	56.3
SVM- SPARROW	8.2 / 88.6	15.0	29.9	49.2	7.2 / 41.3	12.3	21.3	35.1
GRAPHCODEBERT	20.4 / 81.8	32.7	51.1	76.6	23.3 / 80.7	36.1	54.0	66.2
DGCNN	24.3 / 79.3	37.4	54.8	78.7	21.7 / 50.8	26.6	34.70	58.3
GAT	22.8 / 73.6	35.6	52.2	76.7	21.8 / 68.7	31.1	51.27	66.7
GIN	11.1 / 77.1	34.8	52.1	76.9	23.1 / 49.1	33.4	48.95	67.0
GINE	26.8 / 81.6	**40.4**	**57.98**	**80.85**	26.9 / 79.1	**40.1**	**57.01**	**70.06**

Performance & Generality

RQ2:



☐ **TABLE II, Fig.5**

Matches **~67%** of alarms reduced by T, **7.5×** faster, and with **5× less** TO

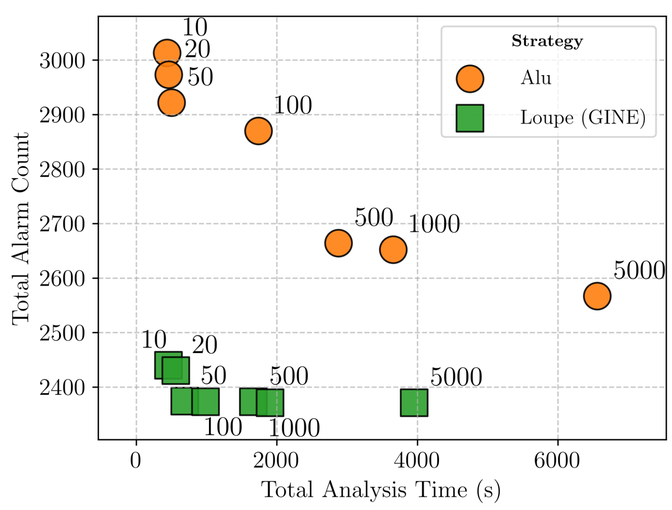
Reduces around **35%** of alarms compared to ⊥

Outperforms or **matches** Alu-100 **87%** of the time with almost **half** the time

Aligns with **51 out of 63** manually labeled unrolling decisions **(81%)**

Consistent performance across **14 abstract domain** settings

☐ **Loupe (GINE) vs Alu with varying unrolling factors – Fig.4**



RQ3:

☐ **TABLE II**

1. Loupe (GINE) trained on Frama-C/Eva **improved** Mopsa performance
2. Retraining on Mopsa's data achieved **75%** of T's alarm reduction
3. The Mopsa model ran **6.5×** faster than T

Key Takeaways

- ✓ **Effective: >25%** improvement in F2 score over feature-engineered methods
- ✓ **Precise & Efficient: Reduces false alarms** by a factor of **1.5** and improves analysis efficiency by **>50%** compared to built-in heuristics
- ✓ **Expert-Level Accuracy:** Matches **>81%** of expert loop unrolling annotations in Open Source Case Studies
- ✓ **Abstract Domains Consistency:** Loupe (GINE) maintains **uniform** alarm-reduction trends and **outperforms** Alu across all abstract domains.
- ✓ **Cross-Analyzer Generalization:** Successfully **transfers** from Frama-C/Eva to Mopsa, demonstrating **robust adaptability**

Maykel Mattar[1,2], Michele Alberti[1], Valentin Perrelle[1], Salah Sadou[2]

1. *Université Paris-Saclay, CEA, List, Palaiseau, France*, 2. *Université Bretagne Sud, IRISA, Vannes, France*

ASE 2025