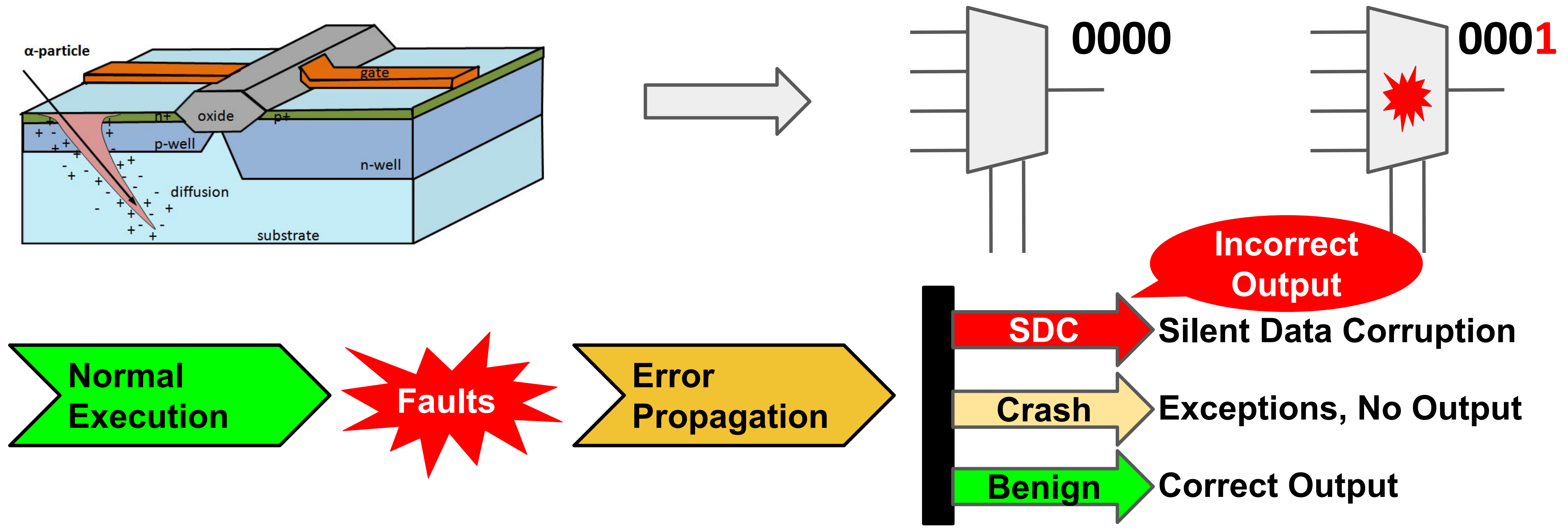
The University of Iowa Argonne National Laboratory Baidu USA

POSTER: Hardening Selective Protection across Multiple Inputs for HPC Applications

Motivation: Soft Errors



0000

0001

Incorrect Output

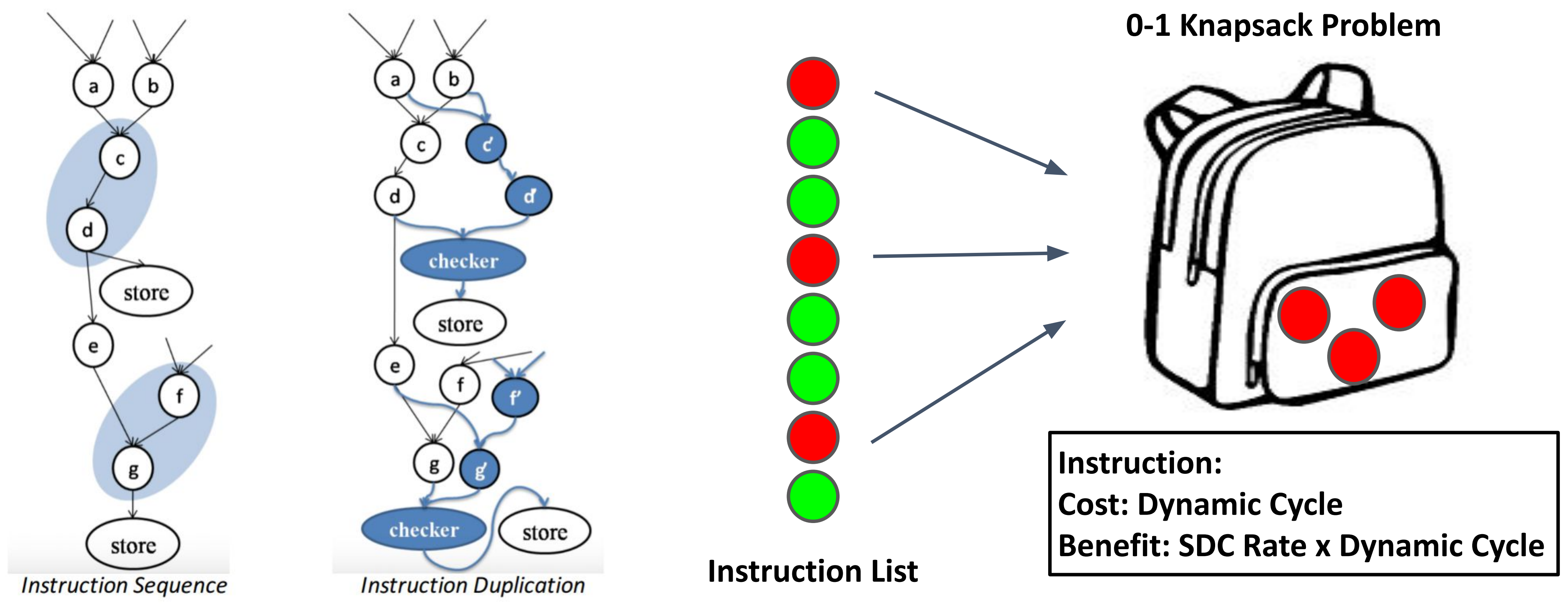
Normal Execution → Faults → Error Propagation

SDC → Silent Data Corruption

Crash → Exceptions, No Output

Benign → Correct Output

Background: Selective Instruction Duplication



Instruction Sequence

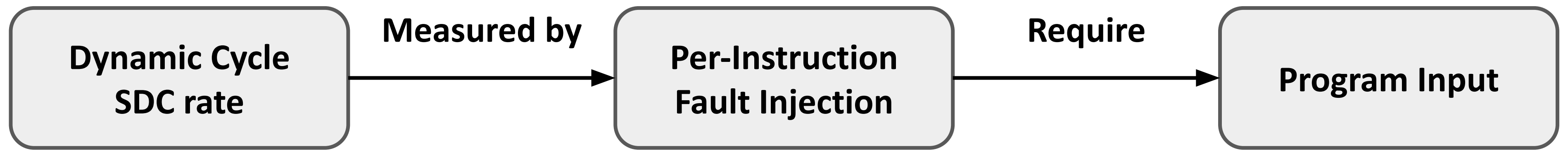
Instruction Duplication

Instruction List

0-1 Knapsack Problem

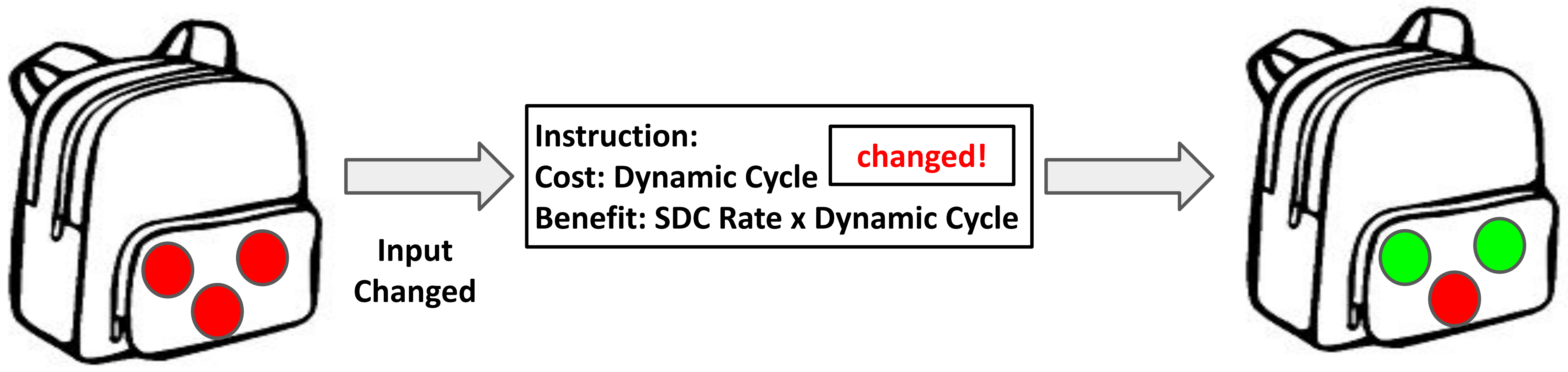
Instruction:
Cost: Dynamic Cycle
Benefit: SDC Rate x Dynamic Cycle

Problem



Dynamic Cycle SDC rate → Measured by → Per-Instruction Fault Injection → Require → Program Input

- When program input changes, per-instruction dynamic cycle and SDC rate will also change!
- Time cost of fault injection is huge! Can not repeat.



Input Changed

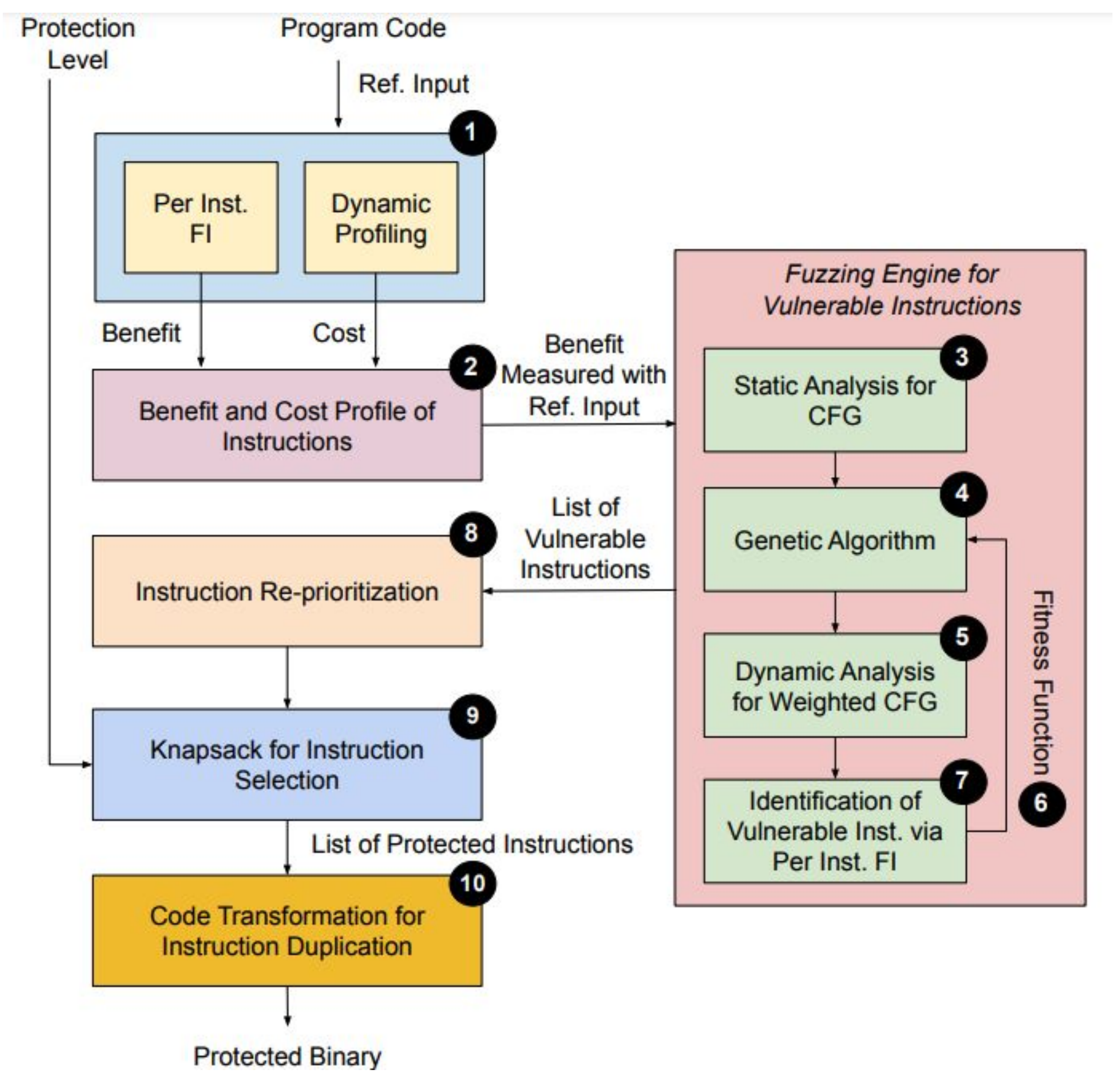
Instruction:
Cost: Dynamic Cycle
Benefit: SDC Rate x Dynamic Cycle

changed!

The protection is not effective anymore!

Solution: SENTINEL

- **Goal:**
  - Protecting programs under arbitrary inputs.
- **Key Insight:**
  - Only a small set of instructions drastically changes benefits while changing inputs. We call them ***vulnerable instructions***.
- **Key Idea of Workflow:**
  - Fuzzing technique to automatically detect *vulnerable instructions*.
  - Instruction re-prioritization. Forcibly protecting ***vulnerable instructions***.
  - Code transformation and compile.



Protection Level

Program Code

Ref. Input

1 Per Inst. FI | Dynamic Profiling

Benefit | Cost

2 Benefit and Cost Profile of Instructions

Benefit Measured with Ref. Input

Fuzzing Engine for Vulnerable Instructions

3 Static Analysis for CFG

4 Genetic Algorithm

5 Dynamic Analysis for Weighted CFG

7 Identification of Vulnerable Inst. via Per Inst. FI

6 Fitness Function

List of Vulnerable Instructions

8 Instruction Re-prioritization

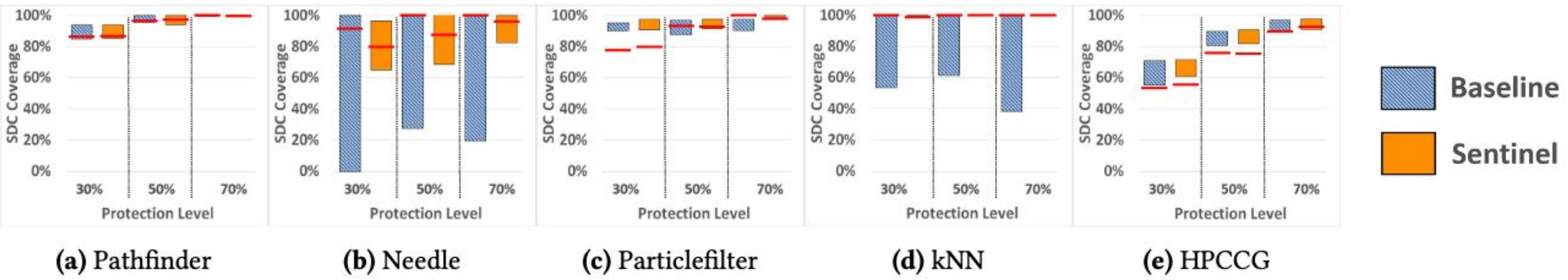
9 Knapsack for Instruction Selection

List of Protected Instructions

10 Code Transformation for Instruction Duplication

Protected Binary

Evaluation



(a) Pathfinder (b) Needle (c) Particlefilter (d) kNN (e) HPCCG

Benchmark	30% Level	50% Level	70% Level
Pathfinder	13.33%	3.33%	3.33%
Needle	3.33%	3.33%	3.33%
Particlefilter	0.00%	56.67%	0.00%
kNN	0.00%	0.00%	0.00%
HPCCG	0.00%	0.00%	3.33%
Average	3.33%	12.67%	2.00%
Average (Baseline)	23.33%	42.67%	44.67%

- SENTINEL can significantly reduce the SDC coverage variation among arbitrary inputs.
- More inputs can reach the expected coverage.

Yafan Huang, Shengjian Guo, Sheng Di, Guanpeng Li, Franck Cappello

yafan-huang@uiowa.edu https://hyfshishen.github.io/