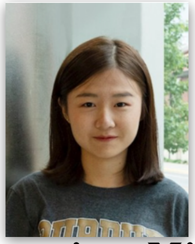
CEDAR: Continuous Testing of Deep Learning Libraries


Danning Xie[1]

Jiannan Wang[1]

Hung Viet Pham[2]

Lin Tan[1]

Yu Guo[3]

Adnan Aziz[3]

Erik Meijer[3]

1 PURDUE UNIVERSITY

2 YORK UNIVERSITÉ UNIVERSITY

3 Meta

Continuous testing for DL libraries is in high demand

PyTorch

TensorFlow

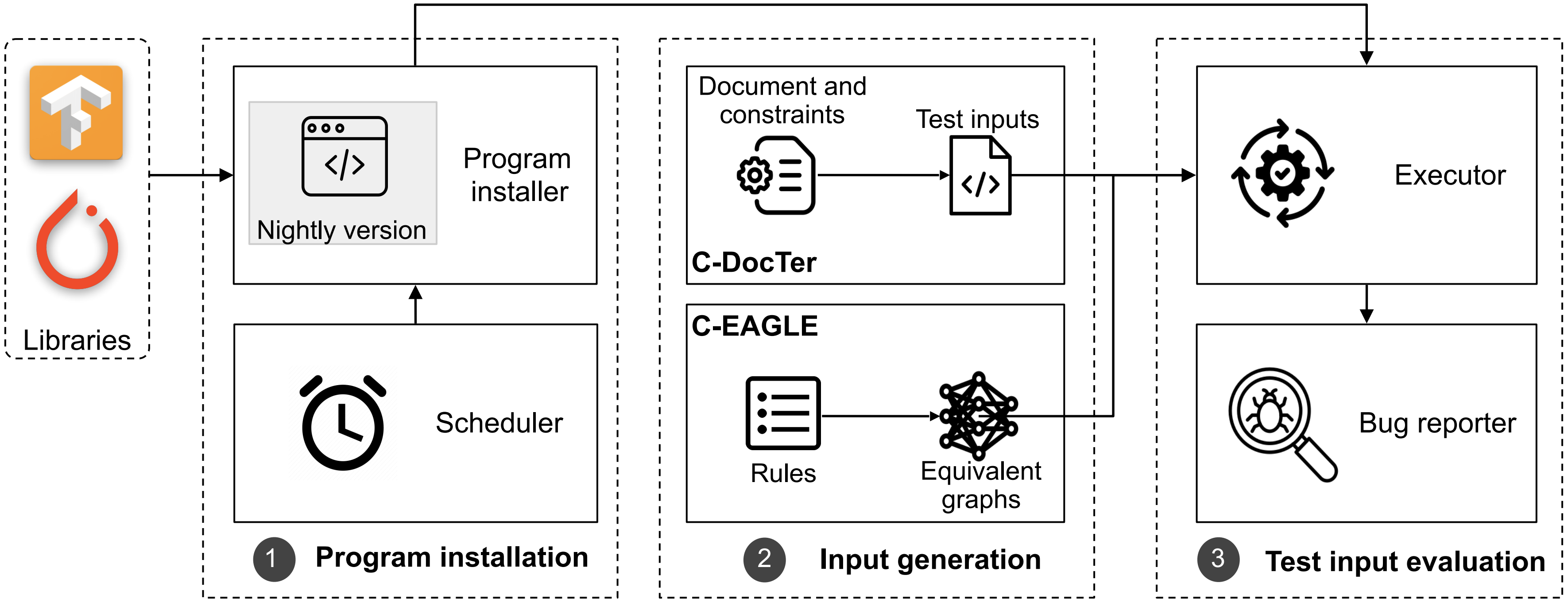
- Nightly build every few days
- 8 official versions in 2023
- 142,385 lines of code changes in a month

- **New bugs** are introduced along with the rapid changes.
- **Existing solutions** do not integrate cutting-edge DL testing tools including:
  - DocTer (ISSTA 22): documentation-guided fuzz testing framework for DL libraries.
  - EAGLE (ICSE 22): differential testing framework with equivalent graphs for DL APIs.
  - …

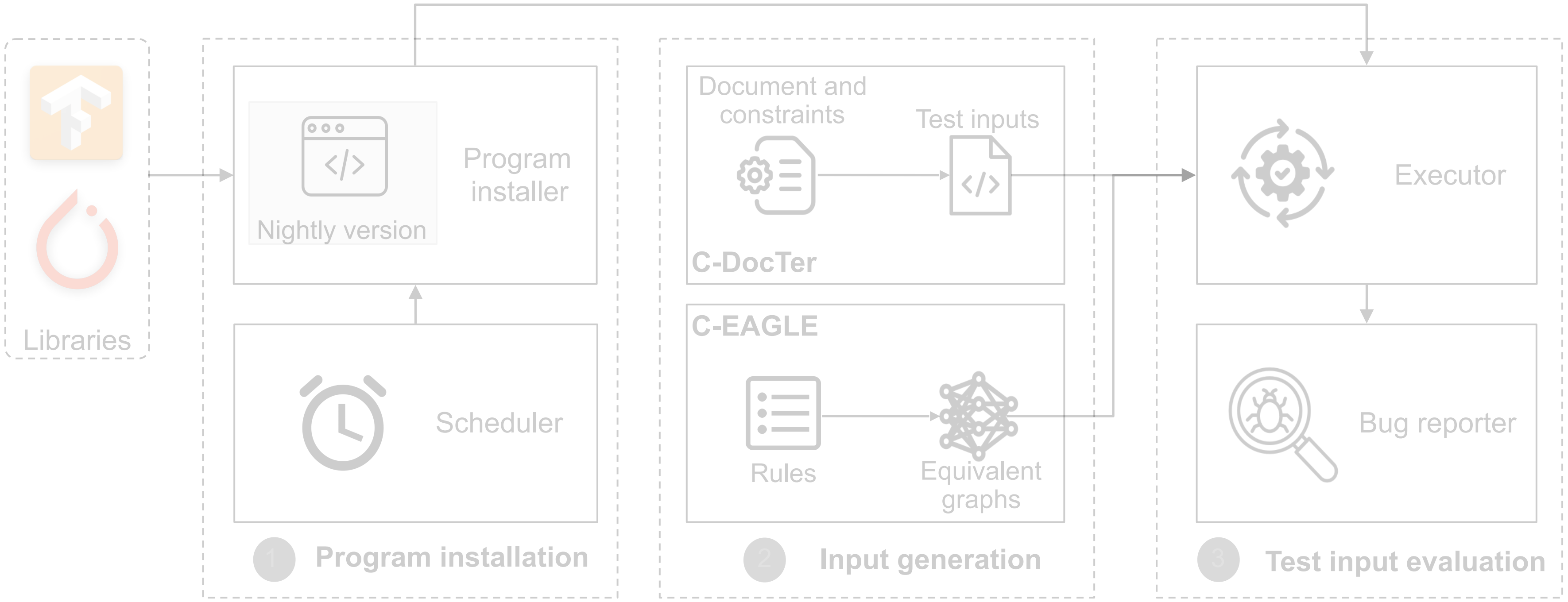
CEDAR: a continuous testing framework

- Integrates two state-of-the-art DL testing approaches (DocTer and EAGLE).
- **Effective**: detecting **83 bugs** affecting 140 PyTorch and TensorFlow APIs, including **23** previously unknown bugs.
- **Efficient**: with tool-specific optimization strategies to reduce the time and space overhead.
- Shortens the bug detection latency by **338.6 days** on average.

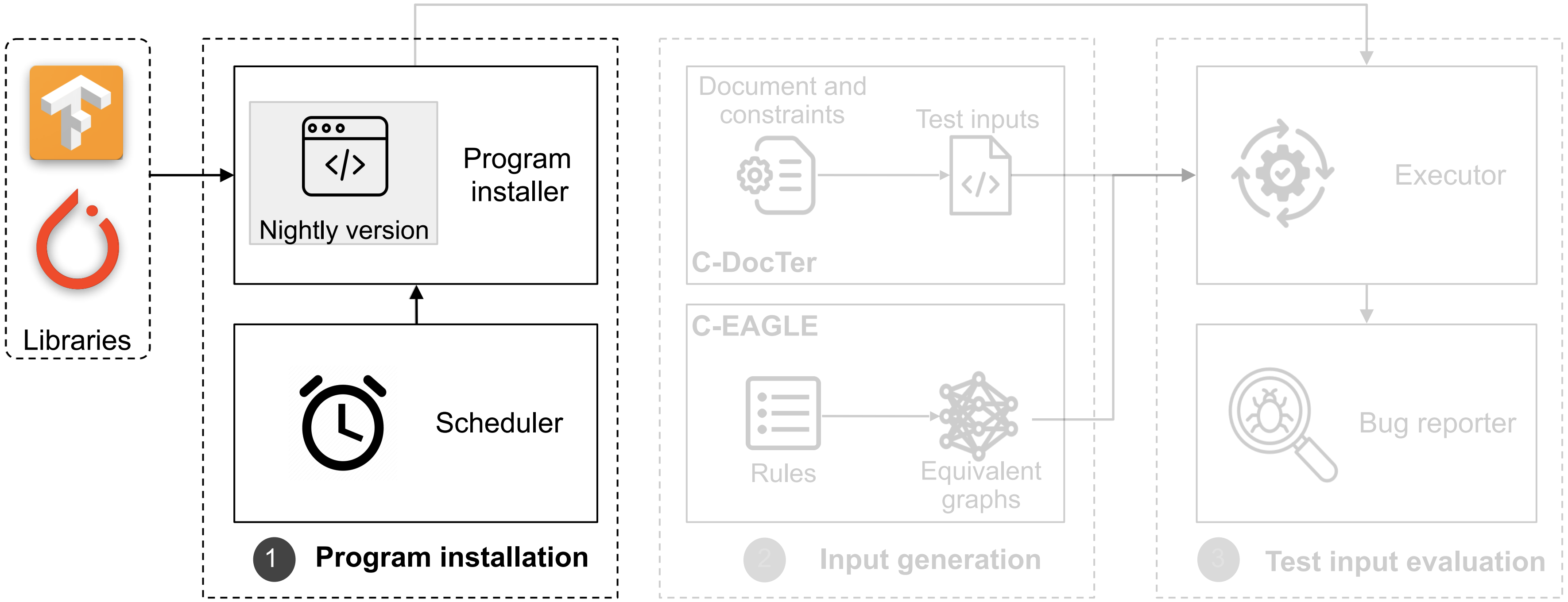
CEDAR overview



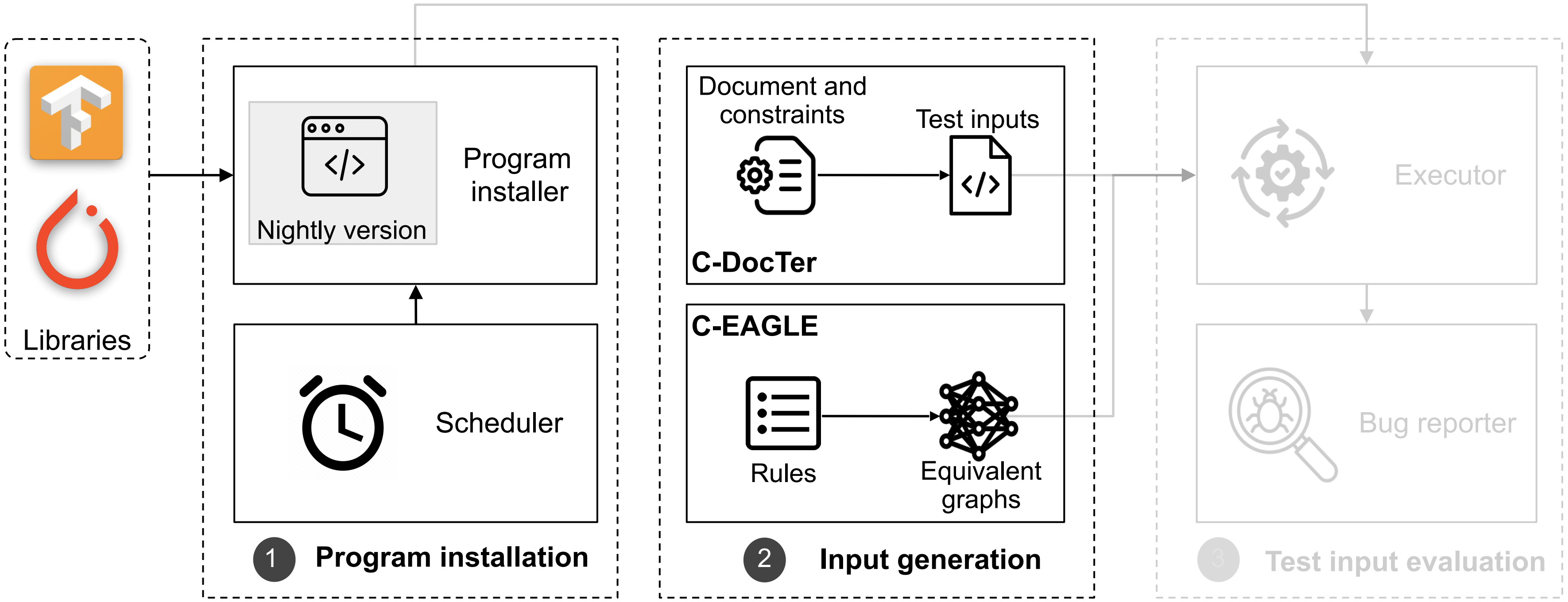
CEDAR overview



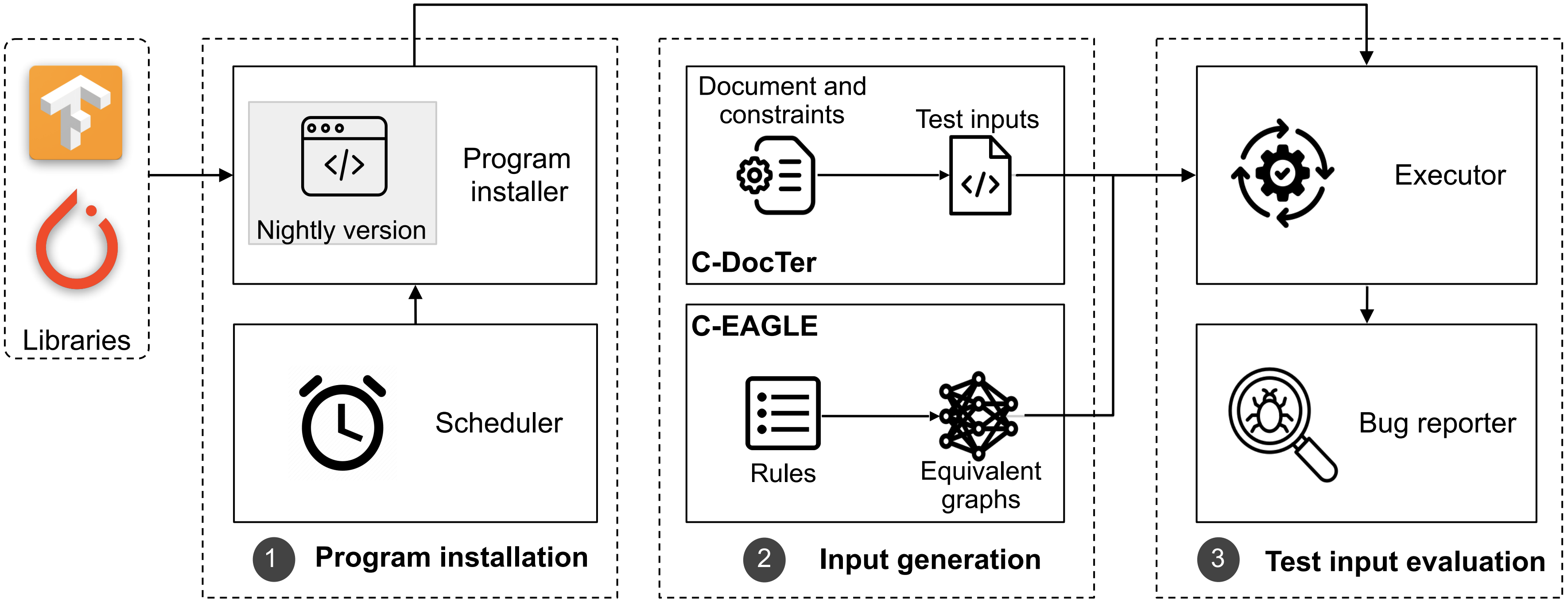
CEDAR overview



CEDAR overview

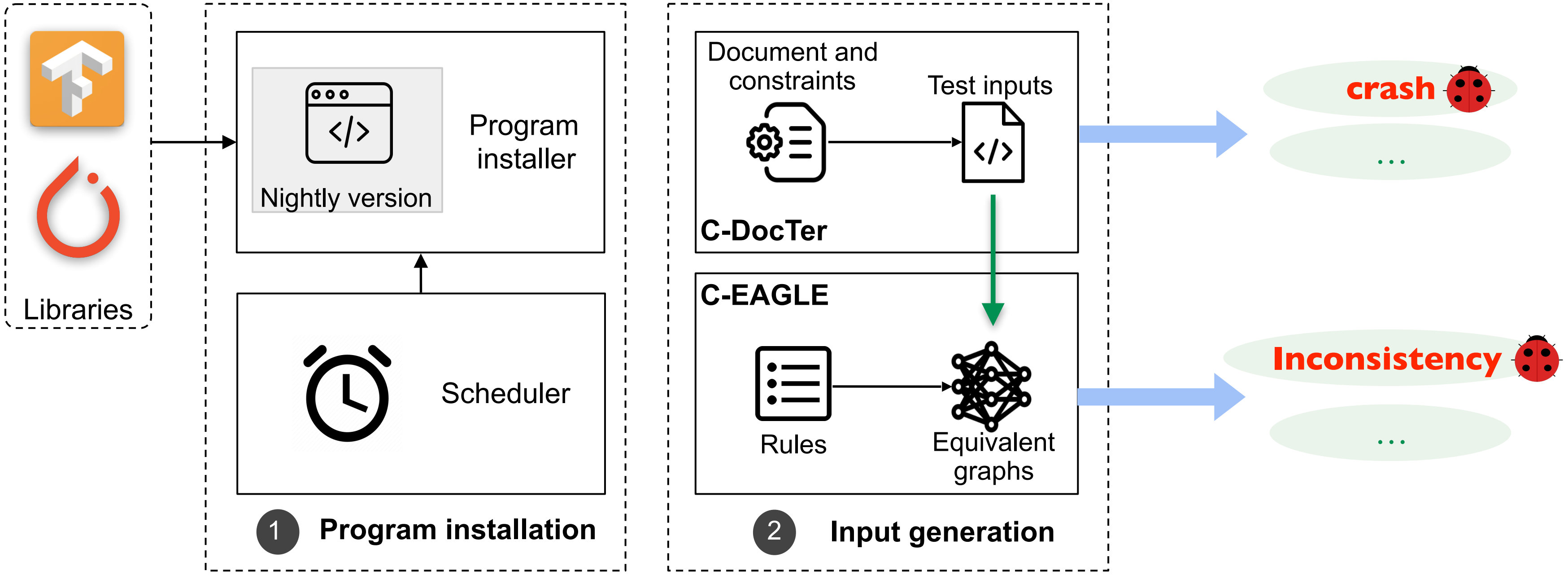


CEDAR overview

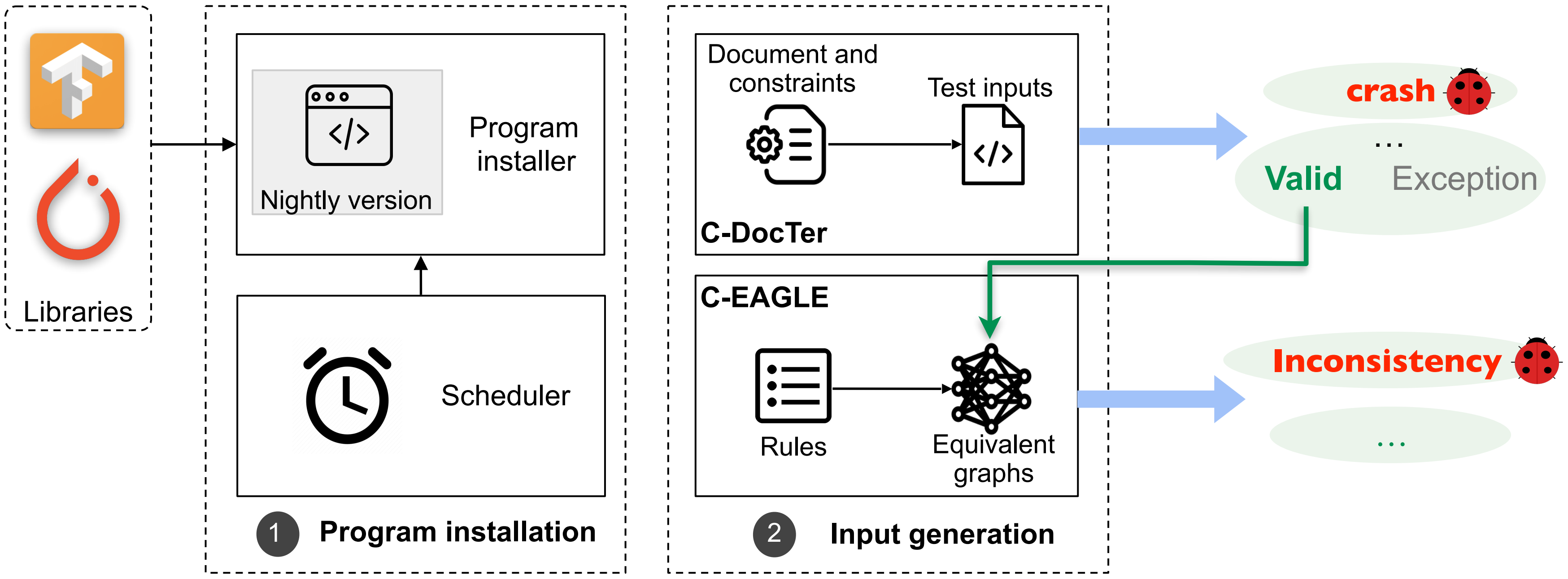


optimization strategies

CEDAR: tool-specific optimization to accelerate input generation



CEDAR: tool-specific optimization to accelerate input generation



Test case reduction

Feed *valid* inputs generated by C-DocTer directly to C-EAGLE

Experimental Setup

2 libraries

10 versions of each library evaluated

519 and 925 APIs from PyTorch and TensorFlow

1,000 test inputs for each API

24 parallel processes

CEDAR's Bug detection results over continuous testing

83 bugs affecting **140** APIs,
including **2** ***high-priority*** bugs (**24** APIs),
23 of the **83** bugs are new,
21 of **23** are confirmed or fixed.

	Verified	New	All	API
PyTorch	6	6	8	35
TensorFlow	15	17	75	105
Total	21	23	**83**	**140**

Bug detection latency reduction

`tf.linalg.diag`

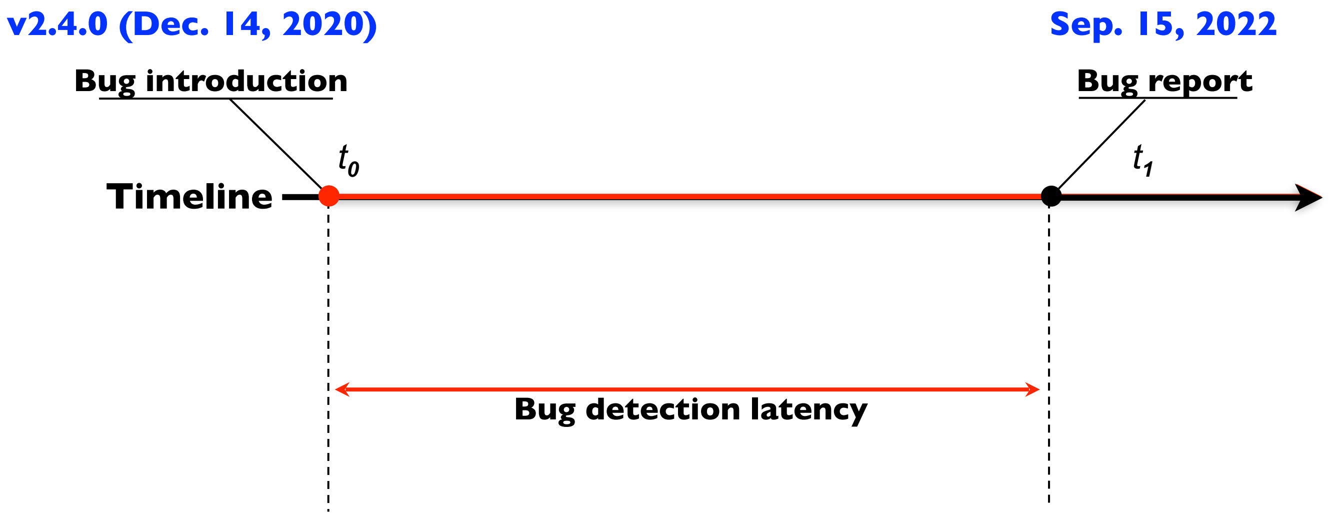
Between ***introduction*** to ***detection***

v2.4.0 (Dec. 14, 2020)
Bug introduction
t_0
Timeline

Bug detection latency reduction

`tf.linalg.diag`

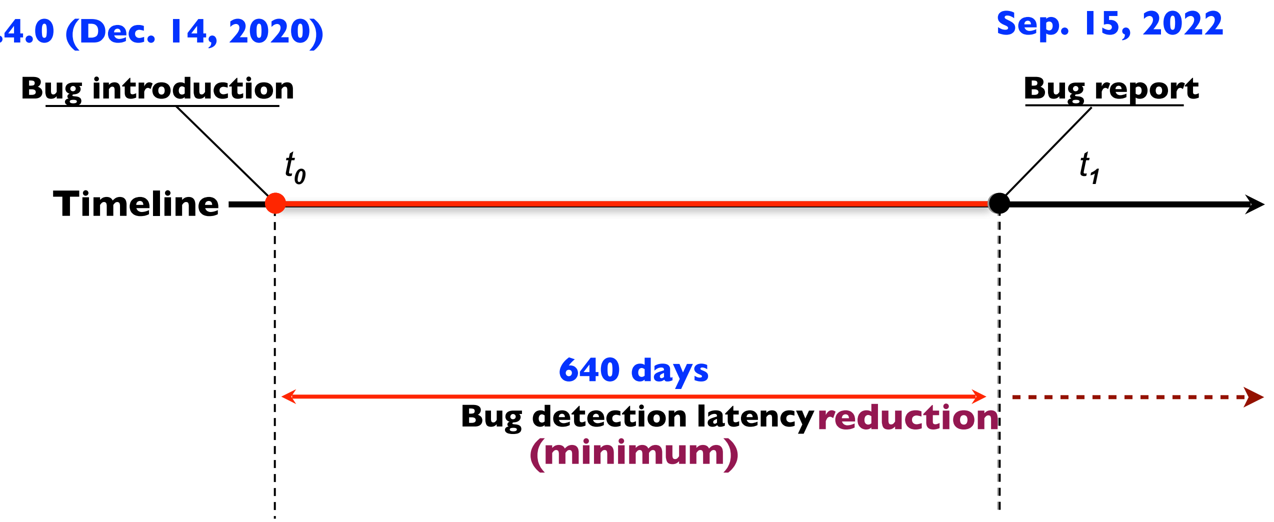
Between ***introduction*** to ***detection***



Bug detection latency reduction

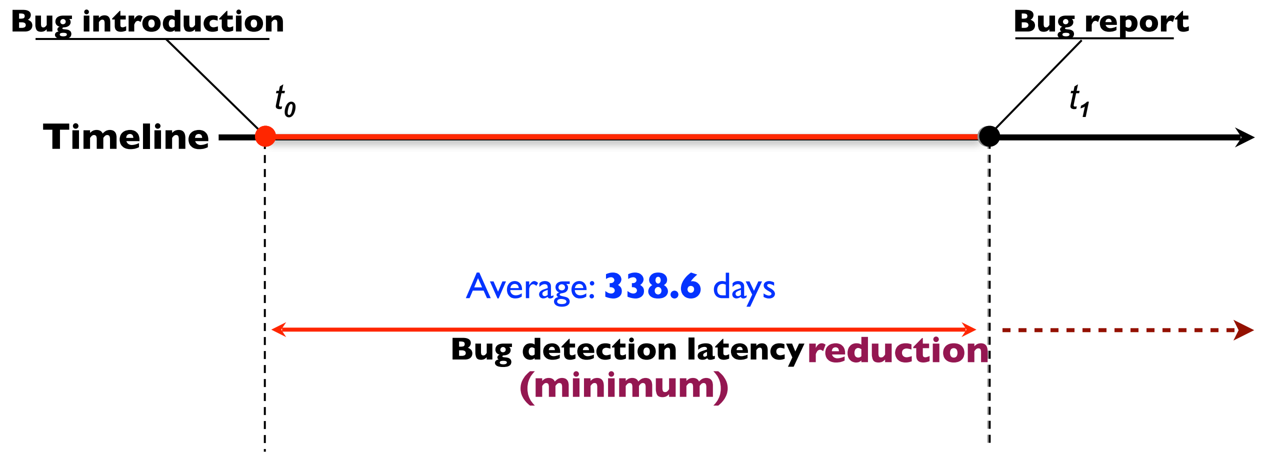
`tf.linalg.diag`

Between ***introduction*** to ***detection***



Bug detection latency reduction

Between ***introduction*** to ***detection***



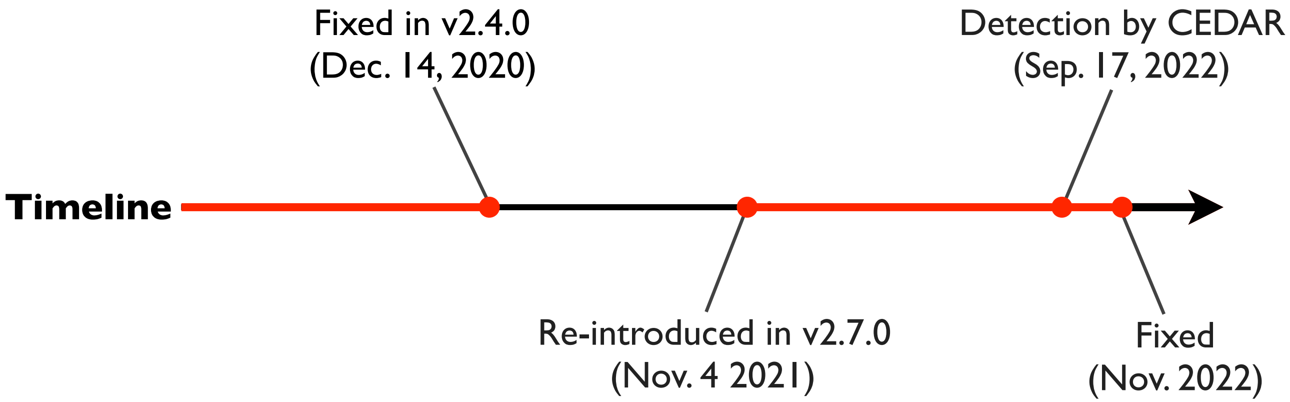
CEDAR leads to bugs being detected at least on average **338.6 days** earlier

CEDAR detected **regression bug** through continuous testing

```
> tf.random.learned_unigram_candidate_sampler(
      true_classes=np.array([[1000000]]),
      num_true=1, …
  )
```

Large value

CRASH! **Segmentation Fault**



CEDAR shows its effectiveness in regression and continuous testing

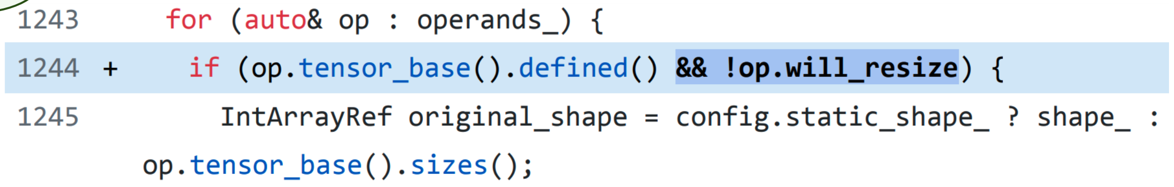
A new **high-priority** bug affecting 23 PyTorch APIs

```
> torch.add(
        input = torch.ones([2,2]),
        other = torch.ones([1]),
        out = torch.ones([2,2,1,1])
  )
```

`out` has at least two more dimensions than both operands

Segmentation Fault

Fix



```
      for (auto& op : operands_) {
+       if (op.tensor_base().defined() && !op.will_resize) {
          IntArrayRef original_shape = config.static_shape_ ? shape_ :
    op.tensor_base().sizes();
```

Reported on Sep. 15th, 2022
Fixed on Sep. 20nd, 2022

A new inconsistency bug

Zero values

original

```
>   o1 = tf.signal.stft(..., frame_length=0, ...)
```

optimized

```
>   @tf.function
    def fun_wrapper(x):
        return tf.signal.stft(*x)
    o2 = fun_wrapper(..., frame_length=0, ...)
```

Reported on May. 9th, 2022
Confirmed on May. 11th, 2022

Large inconsistencies

inconsistency

```
>   np.max(o1 - o2)  # (1.2623837153272947e+180+2.19373012209e-312j)
```

Effectiveness of the optimization strategies

Time efficiency

- shorten the execution time from 130:36 to **8:29**
- reduce the **time overhead** by a factor of **11.3**

Space efficiency

- remove **3,140,929** redundant files in total
- release **159.2 GB** space
- reduce the **space overhead** by a factor of **9.7**

Conclusion and Discussion

- We propose **CEDAR**, a continuous testing framework for DL libraries that efficiently integrates two state-of-the-art DL testing approaches to test DL libraries for detecting bugs effectively.
- CEDAR detected **83 bugs** affecting 140 PyTorch and TensorFlow APIs, including **23 previously unknown** bugs with **21** confirmed or fixed.
- The optimization strategies reduce the time and space overhead by a factor of **11.3** and **9.7**.
- CEDAR's continuous application through 20 versions has effectively shortened the bug detection latency by almost a year (**338.6 days**).