



Self-Supervised Bug Detection and Repair

Miltiadis Allamanis, Henry Jackson-Flux, Marc Brockschmidt

Microsoft Research, Cambridge, UK

Why Self-Supervised Bug Detection and Repair?

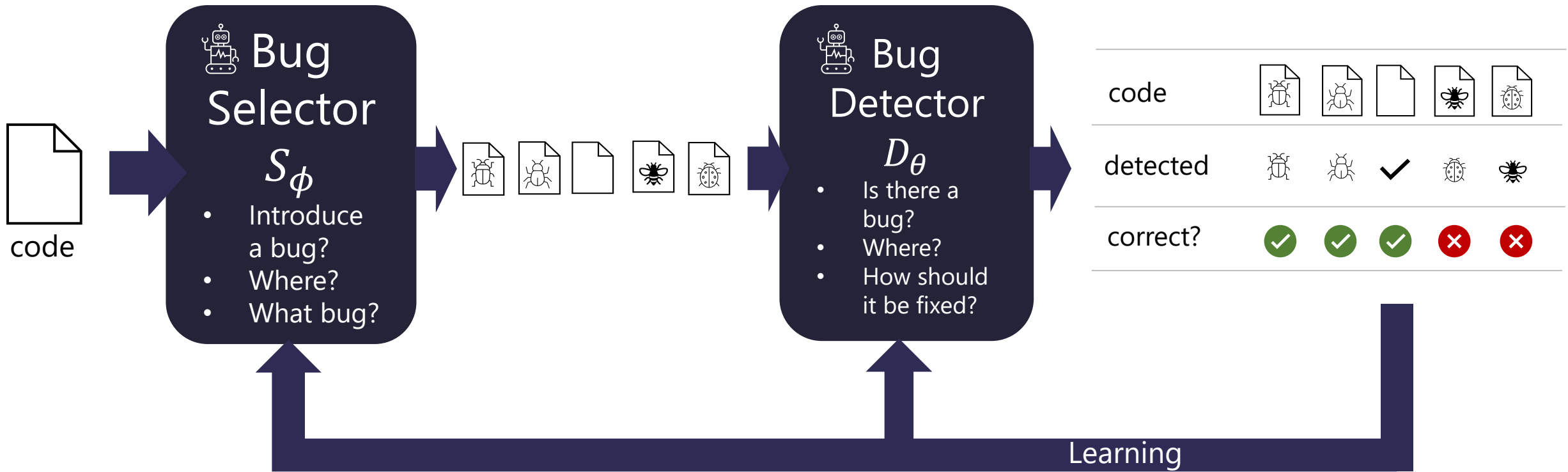
```
def make_id(name):  
    """  
    Create a random id combined with the creditor name.  
    @return string consisting of name (truncated at 22 chars), -,  
    12 char rand hex string.  
    """  
  
    r = get_rand_string(12)  
    if len(name) <= 22:  
        name = name[:22]  
    return name + "-" + r
```

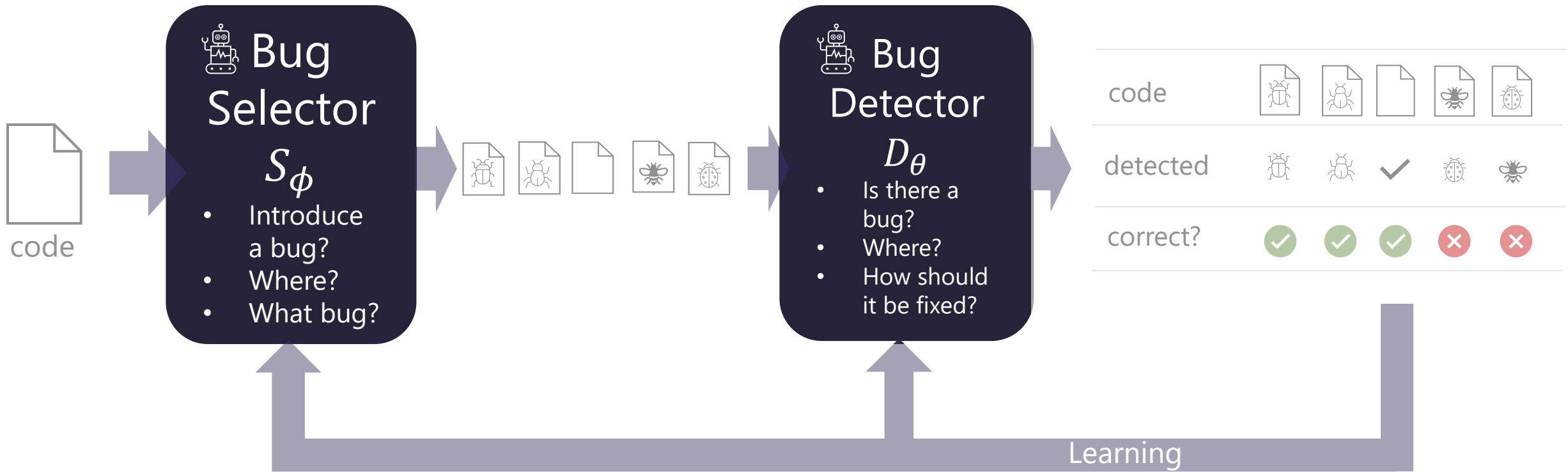
<https://github.com/raphaelm/python-sepaxml.git:/sepadd/utils.py>

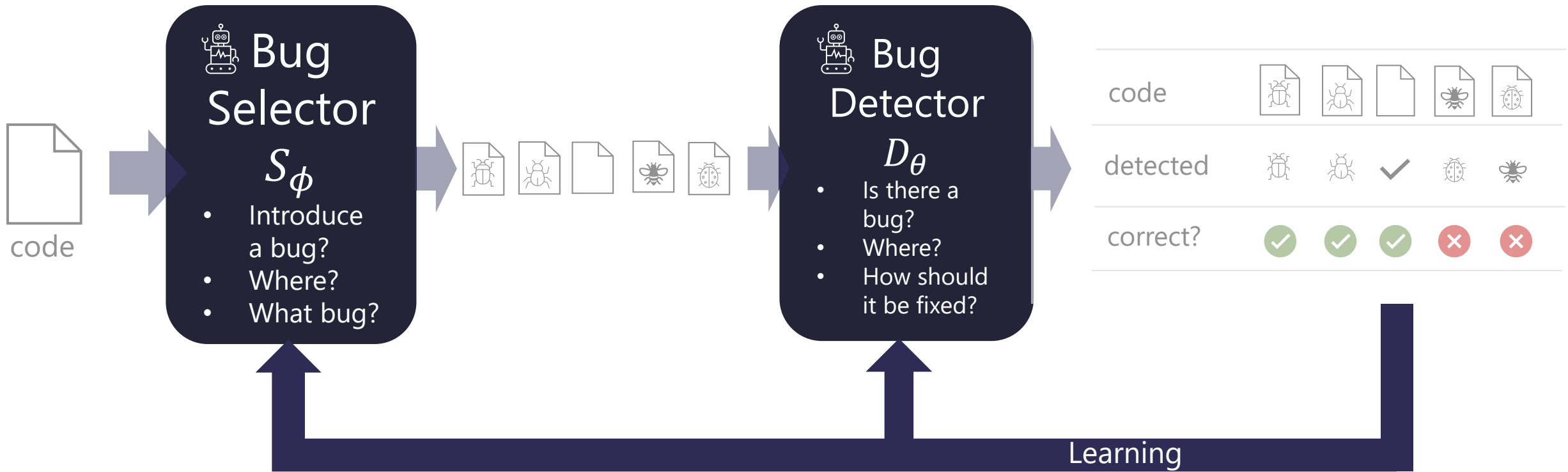
Why Self-Supervised Bug Detection and Repair?

```
def make_id(name):  
    """  
    Create a random id combined with the creditor name.  
    @return string consisting of name (truncated at 22 chars), -,  
    12 char rand hex string.  
    """  
  
    r = get_rand_string(12)  
    if len(name) <= 22:  
        name = name[:22]  
    return name + "-" + r
```

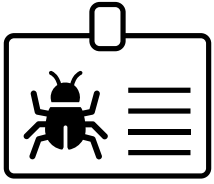
<https://github.com/raphaelm/python-sepaxml.git:/sepadd/utils.py>







$$\max_{\phi} \min_{\theta} E_{s \sim C} [E_{\langle l, \rho \rangle \sim S_{\phi}(s)} [\mathcal{L}_{D_{\theta}}(s[\rho]l, \langle l, \rho^{-1} \rangle)]]$$



Types of Rewrites

- Replace Variable Usage
- Replace Binary Operator
- Replace Assignment Op
- Replace Boolean Operator
- Replace Comparison Operator
- Replace (some) Literals
- Argument Swap

Example

`i` → `j`

`+` → `-`

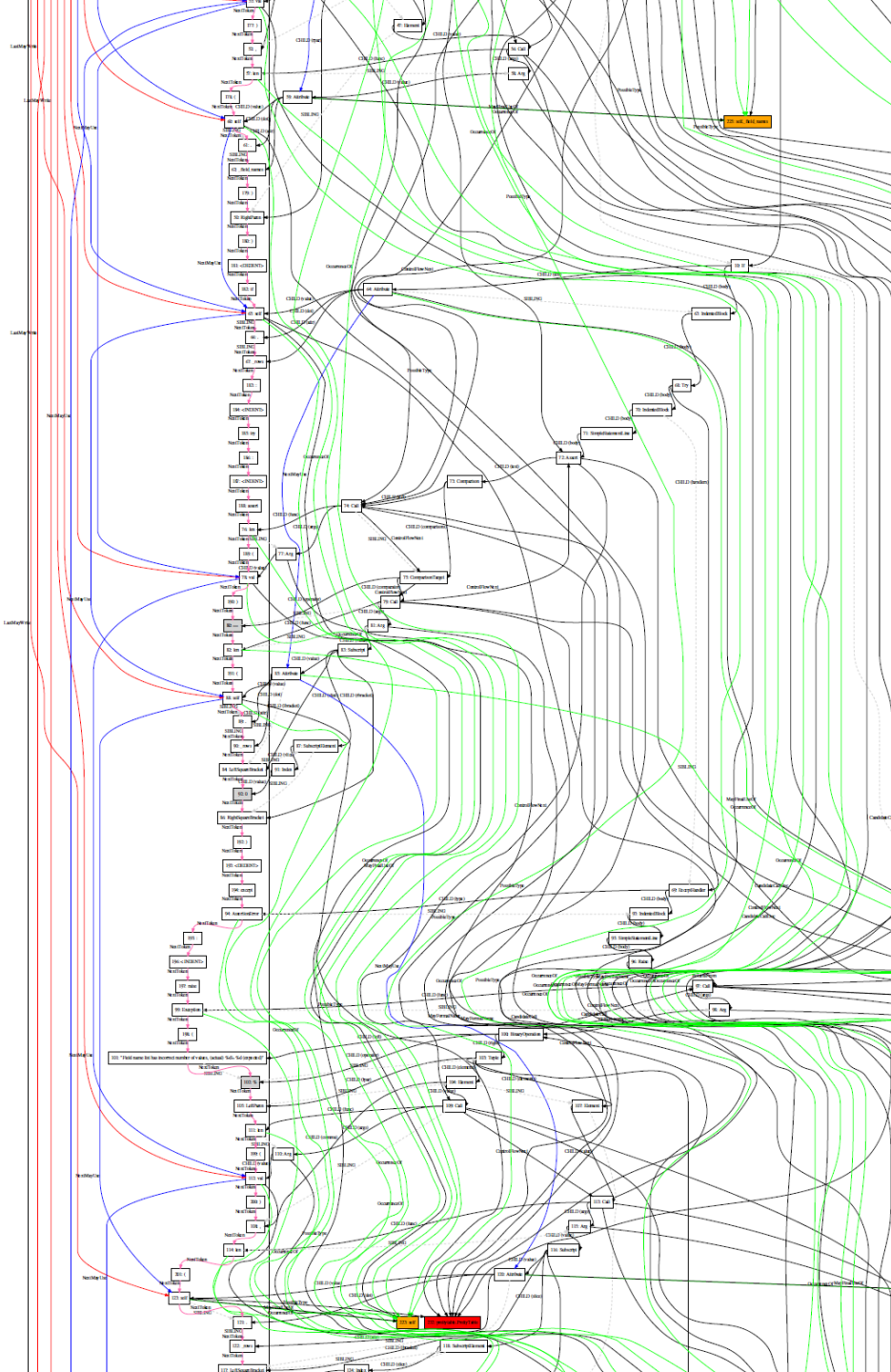
`+=` → `--`

`or` → `and`

`==` → `!=`

`0` → `1`

`foo(a+1, b)` → `foo(b, a+1)`



Code Representation

Entities (Nodes)

- Tokens
- Non-Terminal Nodes
- Symbols

Relationships (Edges)

Syntax

- AST Child
- AST Sibling
- Next Token

Function Calls

- CandidateFormalArg
- CandidateDocStringOf

Data Flow

- MayFinalUseOf
- LastMayWrite
- NextMayUse

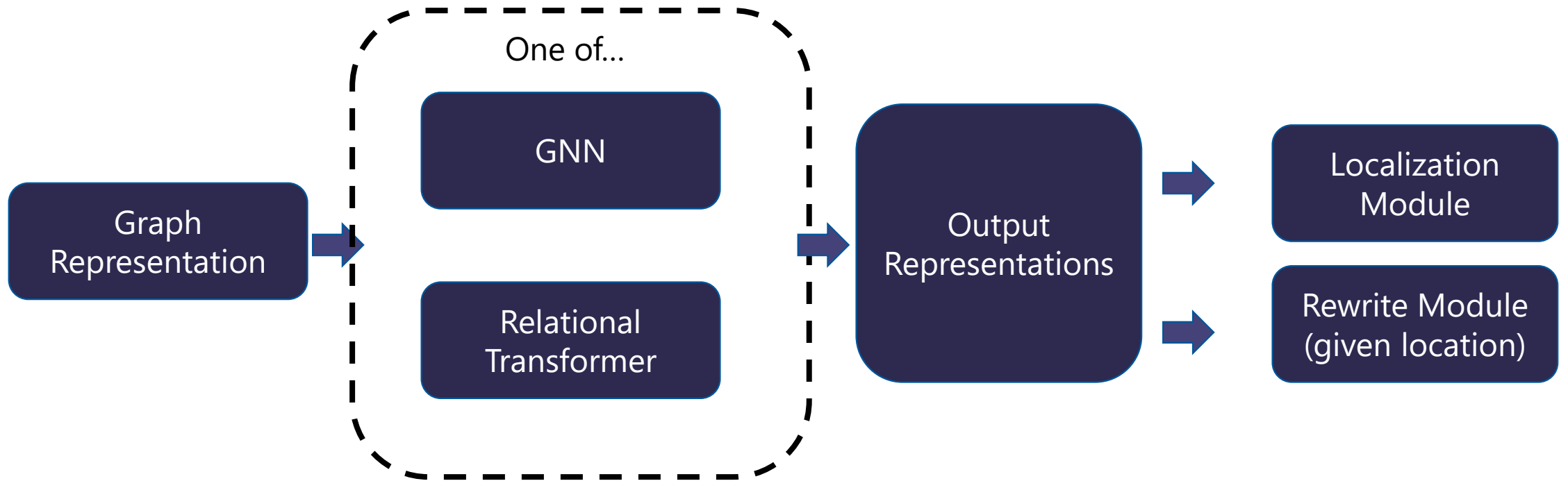
Control Flow

- ControlFlowNext
- AssignedFrom
- ReturnsFrom
- YieldsFrom

Symbols

- CandidateType
- OccurrenceOf
- CandidateMethodName

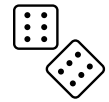
Neural Models



GNN: Allamanis, M., et al. "Learning to Represent Programs with Graphs." *ICLR* 2017

GREAT: Hellendoorn, V. J., et al. "Global Relational Models of Source Code." *ICLR* 2019

Evaluation Datasets



RandomBugs

- ~700k random bugs
- Relatively Large
- Potentially non-representative of real bugs



PyPIBugs

- 2k real bugs
- Manually curated/labeled
- Small. Used as testset only.

Localization & Repair Accuracy



RandomBugs

	GNN	GREAT
Supervised	62.4	51.0
BugLab	70.3	65.3



PyPIBugs

	GNN	GREAT
Supervised	20.1	16.5
BugLab	26.2	22.9

GNN: Allamanis, M., et al. "Learning to Represent Programs with Graphs." *ICLR* 2017

GREAT: Hellendoorn, V. J., et al. "Global relational models of source code." *ICLR* 2019

Conversation 1

Commits 1

Checks 0

Files changed 1

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ ⚙ ▾

2 gremlin-python/src/main/python/gremlin_python/process/strategies.py

↑

@@ -64,7 +64,7 @@ def __init__(self, partition_key=None, write_partition=None, read_partitions=None

64 64 self.configuration["partitionKey"] = partition_key

65 65 if write_partition is not None:

66 66 self.configuration["writePartition"] = write_partition

67 - if write_partition is not None:

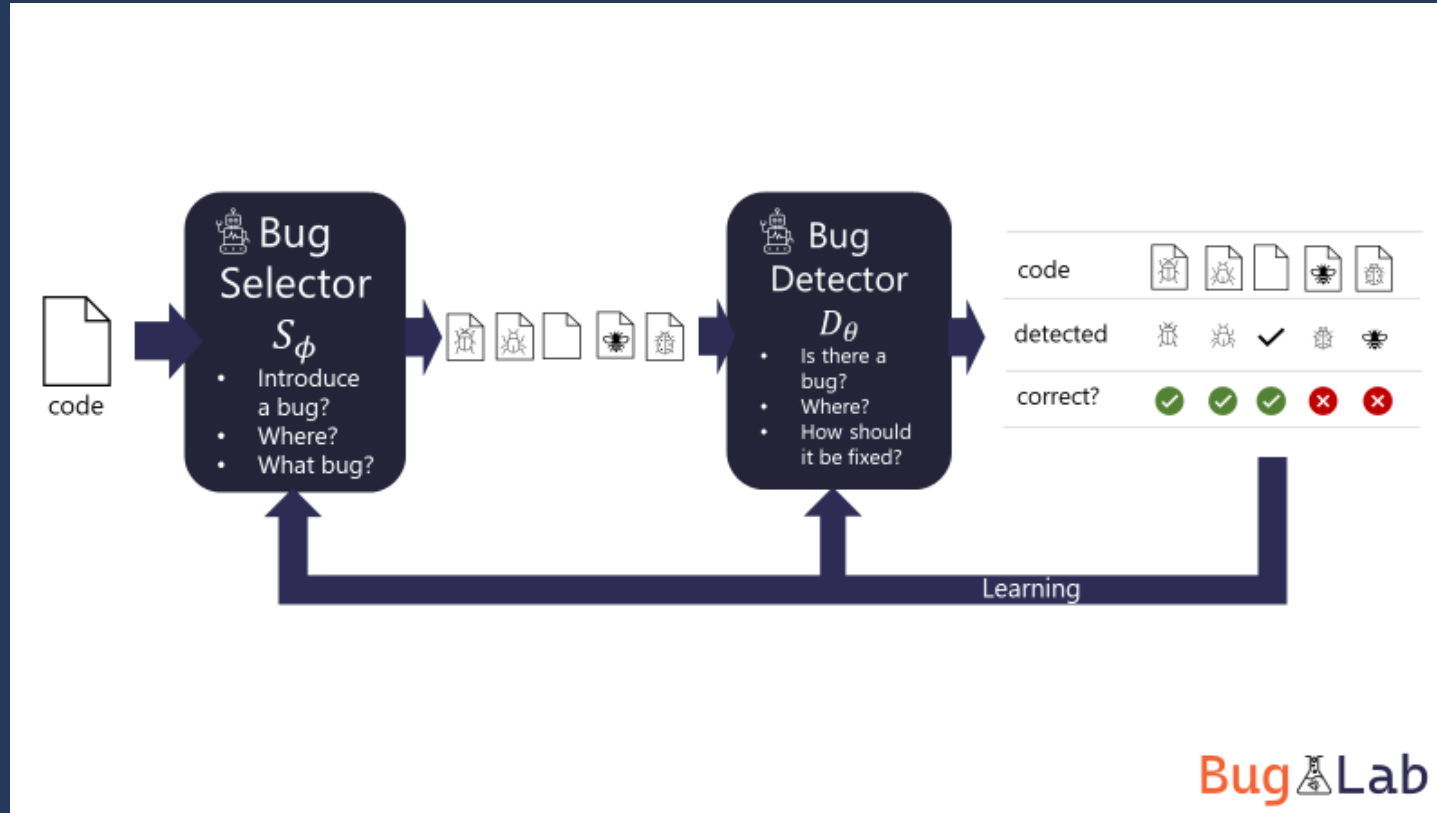
67 + if read_partitions is not None:

68 68 self.configuration["readPartitions"] = read_partitions

69 69 if include_meta_properties is not None:

70 70 self.configuration["includeMetaProperties"] = include_meta_properties

↓



Self-Supervised Bug Detection and Repair

Miltiadis Allamanis, Henry Jackson-Flux, Marc Brockschmidt