

REPAIRING CODE WITH MACHINE LEARNING

MILTOS ALLAMANIS



miltos.allamanis.com



[@miltos1](https://twitter.com/miltos1)

```
import passport from 'passport';
import LocalStrategy from 'passport-local';
import { Strategy as JWTStrategy, Extractor } from 'passport-jwt';
import request from 'superagent';

import user from './models/user-model';
import constants from './config/constants';
import { createToken } from './helpers/auth-helper';

/**
 * Local Strategy Auth
 */
const localOpts = { usernameField: 'username' };
const localLogin = new LocalStrategy(
  localOpts,
  async ({ username, password, done }) => {
    try {
      const user = await User.query().where('username', username);

      if (user.length === 0) {
        const userData = {
          username,
          password,
        };
        const createdAt = await createUser(userData);
        return done(null, createdAt);
      } else if (user[0].authenticate(password) {
        return done(null, false);
      }

      return done(null, user[0]);
    } catch (e) {
      return done(null, false);
    }
  }
);

/**
 * JWT Strategy Auth
 */
const jwtOpts = {
  // Telling Passport to check authorization headers for JWT
  jwtFromRequest: Extractor.fromAuthHeaderWithScheme('JWT'),
  // Telling Passport where to find the secret
  secretOrKey: constants.JWT_SECRET,
};

const jwtLogin = new JWTStrategy(jwtOpts, async ({ payload, done }) => {
  try {
    console.log(payload);
    const user = await User.query().where('user_uid', payload.user_uid);
    console.log(user[0].toString());

    if (user.length === 0 || !user) {
      return done(null, false);
    }

    return done(null, user[0]);
  } catch (e) {
    console.log(e);
    return done(null, false);
  }
});

// .vscode/settings.json
{
  "name": "miltos",
  "version": "1.0.0",
  "description": "A JavaScript IDE",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/miltos1/miltos"
  },
  "keywords": [
    "javascript",
    "ide",
    "miltos"
  ],
  "author": "Miltos Allamanis",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/miltos1/miltos/issues"
  },
  "homepage": "https://github.com/miltos1/miltos"
}
```

```
public static Task<T> Create(int count, out SyncPoint[] syncPoints)
{
    // Create local sync points
    var localSyncPoints = new SyncPoint[count];
    for (var i = 0; i < count; i++)
    {
        localSyncPoints[i] = new SyncPoint();
    }

    syncPoints = localSyncPoints;

    var counter = 0;
    return () =>
    {
        if (counter >= localSyncPoints.Length)
        {
            return Task.CompletedTask;
        }
        else
        {
            var syncPoint = localSyncPoints[counter];

            counter++;
            return syncPoint.WaitToContinue();
        }
    }
}
```

```
2 String username;  
3 String password;  
4  
5  
6  
7  
8 password = username;  
|
```



THE ML4CODE LANDSCAPE

Code
Generation

Program Analysis

...

Code
Completion

Program
Synthesis

Specification
Tuning

Specification
Inference

Black-Box
Analysis
Learning

LEARNED PROGRAM ANALYSES

Specification Tuning & Filtering

- A formal program analysis.
- Tune (discount some factors) to reduce false positives.

Specification Inference

- Assume most code complies with a latent spec.
- Predict a spec.
- Verify with standard methods.

Black-Box Analysis Learning

- Assume most code is “correct”.
- Model (latent) user intent and deviations from it.
- Raise warnings on detected deviations.



DETECTING & REPAIRING BUGS

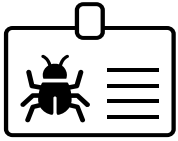
- 📄 Self-Supervised Bug Detection and Repair. *Allamanis, Flux, Brockschmidt. NeurIPS 2021*
- 📄 Learning to Represent Programs with Graphs. *Allamanis, Brockschmidt, Khademi. ICLR 2018*

```
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

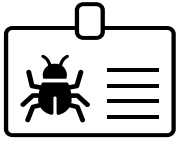


Hard to detect with general hand-written rule or frequent pattern mining



TYPES OF REWRITES

	Example	
Replace Variable Usage	<code>i</code>	<code>→ j</code>
Replace Binary Operator	<code>+</code>	<code>→ -</code>
Replace Assignment Op	<code>+=</code>	<code>→ -=</code>
Replace Boolean Operator	<code>or</code>	<code>→ and</code>
Replace Comparison Operator	<code>==</code>	<code>→ !=</code>
Replace (some) Literals	<code>0</code>	<code>→ 1</code>
Argument Swap	<code>foo(a+1, b)</code>	<code>→ foo(b, a+1)</code>



TYPES OF REWRITES

```
def foo(a, b, c=0):  
    if a[1] in[2] b[3]:  
        c[4] +=[5] bar(b[7], c[8])[6]  
    c_is_neg =[9] c[10] <[11] 0[12]  
    if c_is_neg[13] or[14] a[15] is[16] int:  
        return True[17], c[18]  
    return c[19] >[20] 1[21], c[22]
```

ϵ : NOBUG

l_1 : b, c

l_2 : not in

l_3 : a, c

l_4 : a, b

l_5 : =, -=, *=, /=, //, %

l_6 : bar(c, b)

l_7 : a, c

l_8 : a, b

l_9 : +=, -=, *=, /=, //, %

l_{10} : a, b

l_{11} : <=, >, >=, ==, !=

l_{12} : -2, -1, 1, 2

l_{13} : a, b, c, not c_is_neg

l_{14} : and

l_{15} : b, c, c_is_neg

l_{16} : is not

l_{17} : False

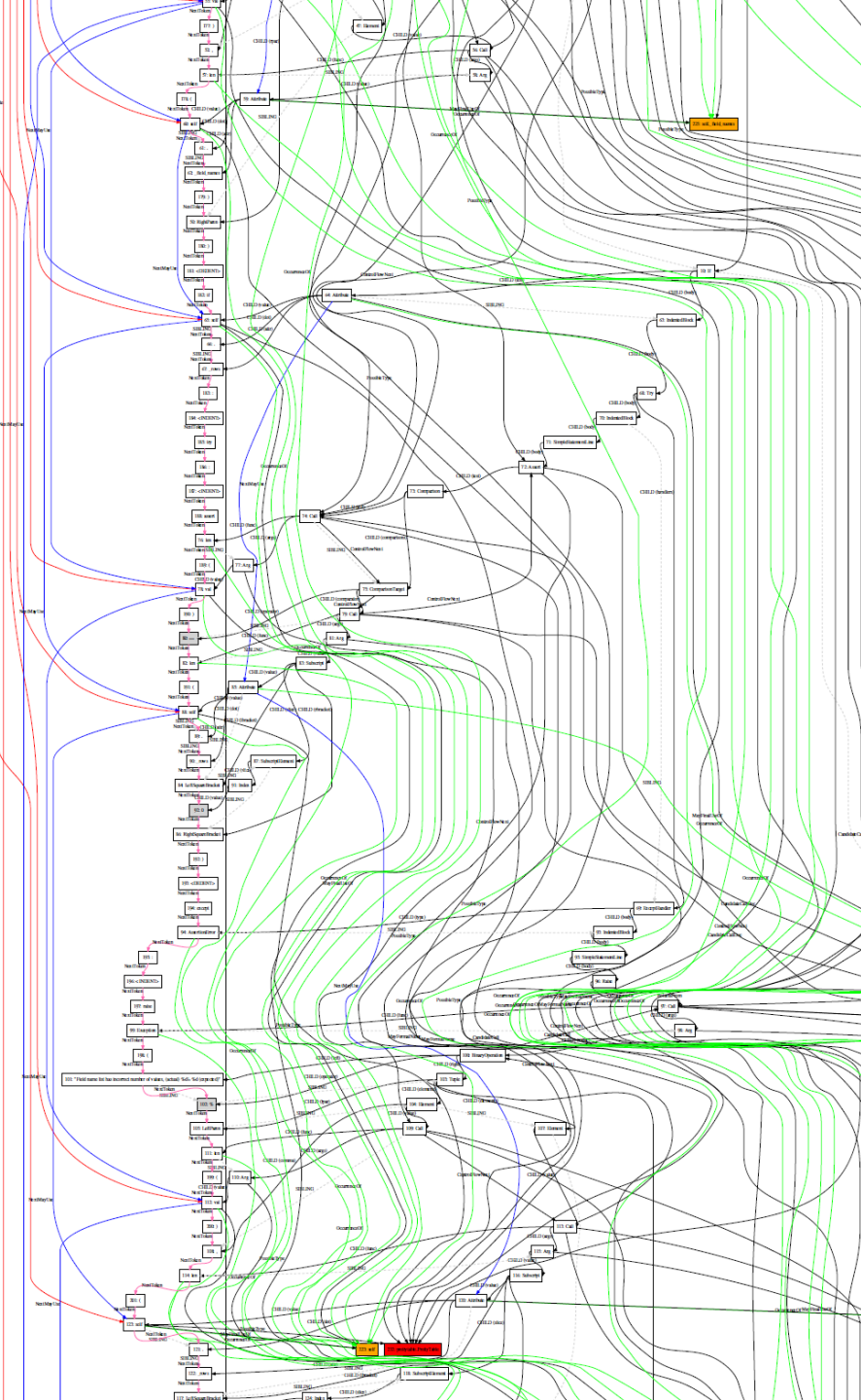
l_{18} : a, b, c_is_neg

l_{19} : a, b, c_is_neg

l_{20} : >=, <, <=, ==, !=

l_{21} : -2, -1, 0, 2

l_{22} : a, b, c_is_neg



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

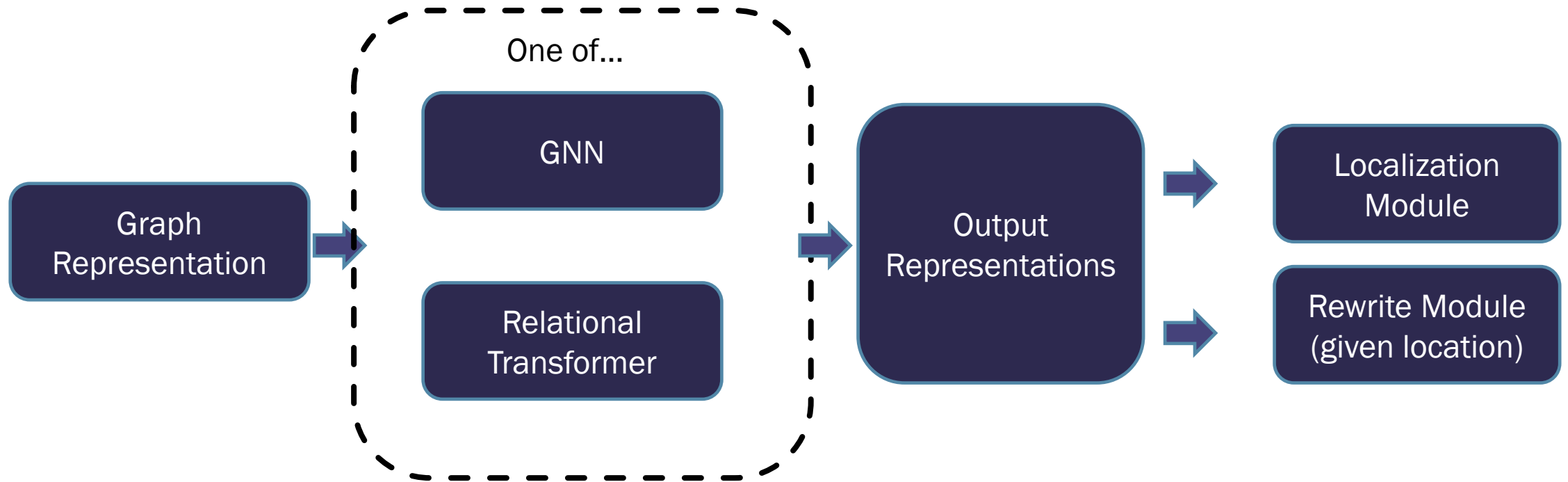
Control Flow

- ControlFlowNext
- AssignedFrom
- ReturnsFrom
- YieldsFrom

Symbols

- CandidateType
- OccurrenceOf
- CandidateMethodName

NEURAL MODELS

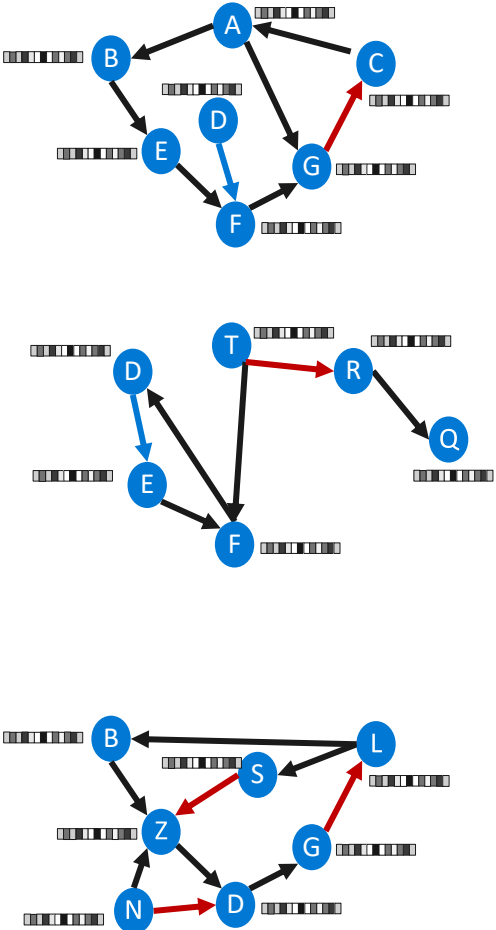


GNN: Allamanis, M., et al. "Learning to Represent Programs with Graphs." *ICLR* 2017

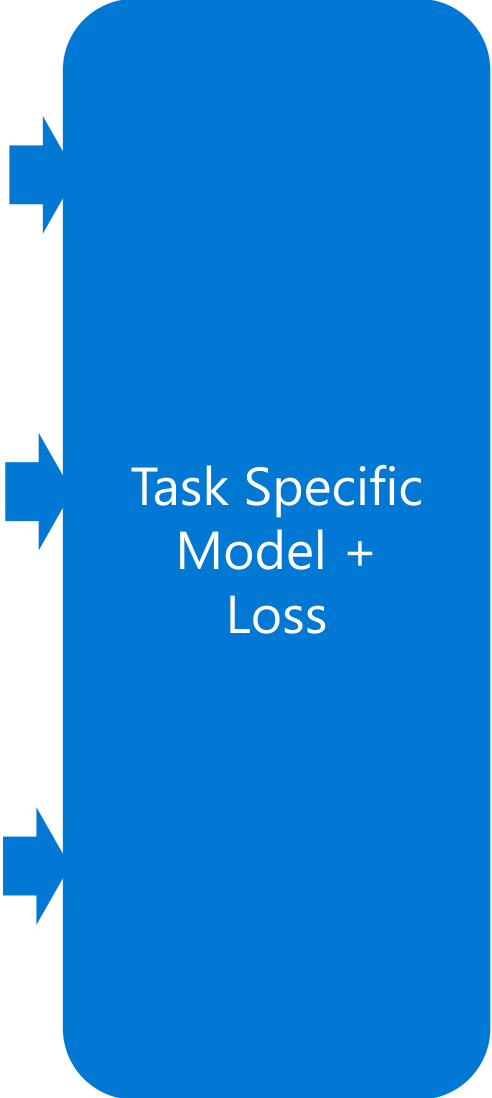
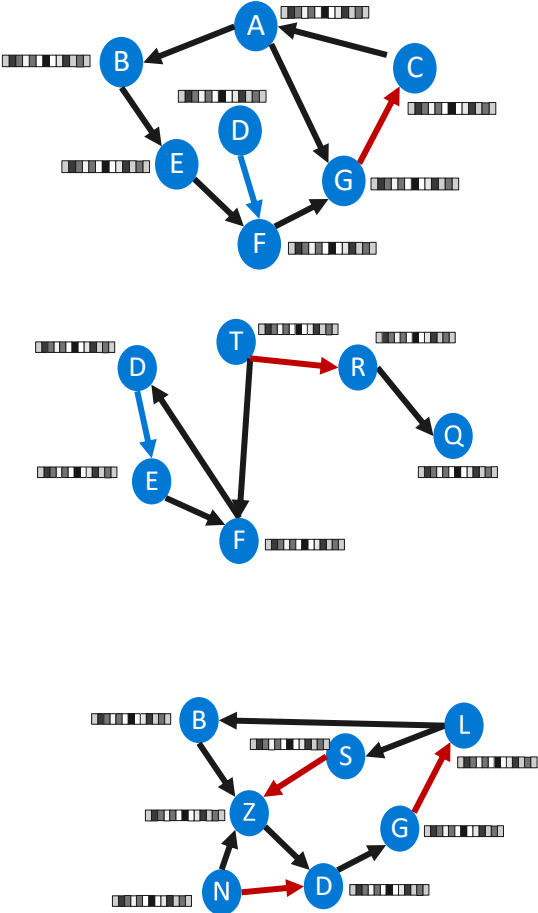
GREAT: Hellendoorn, V. J., et al. "Global Relational Models of Source Code." *ICLR* 2019

Graph Neural Networks in One Slide

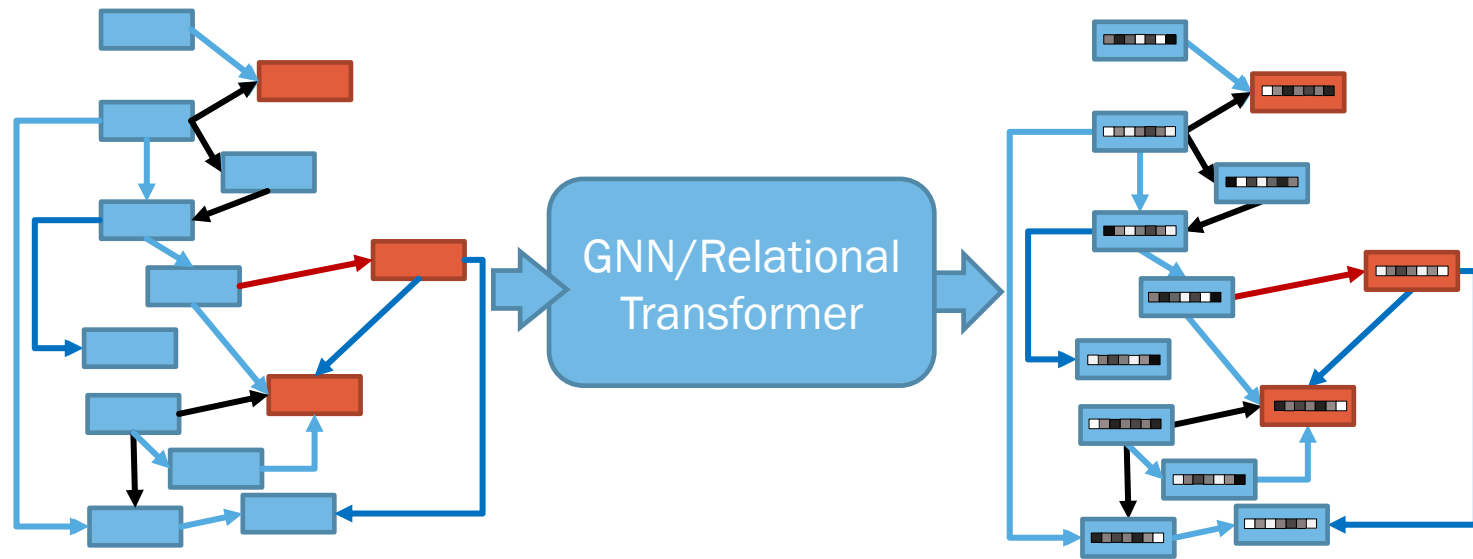
Input Representations



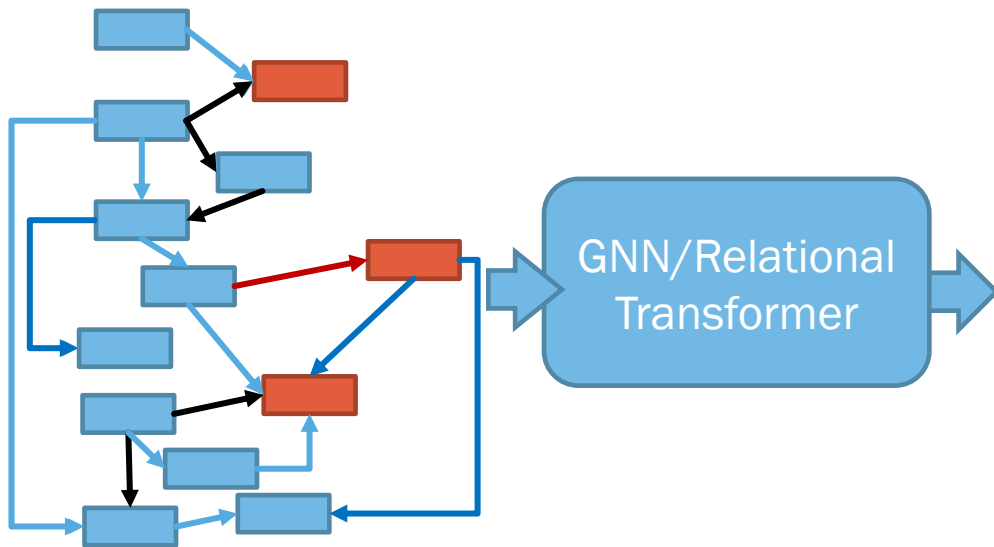
Output Representations



NEURAL ARCHITECTURE



NEURAL ARCHITECTURE



Localization

PointerNet(CandidateNodes...)



Repair Given Location

Variable Misuse

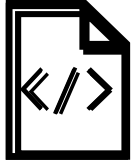
PointerNet(AlternativeNodes... |)

Text Rewrite

MaskedMLP() -> {+, -, *, /, 0, 1, 2, and, or, ...}

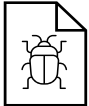
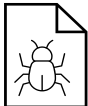
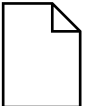






Argument Swapping

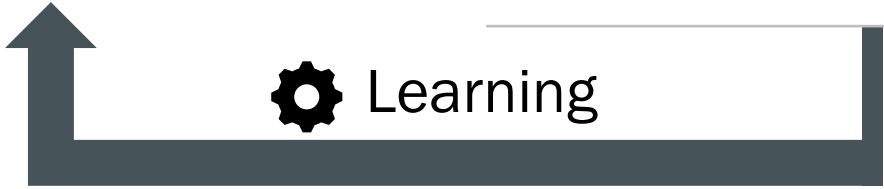
PointerNetOverPairs(ArgNodes... |)



Insert
Random
Bug

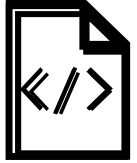
Bug
Detector

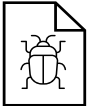
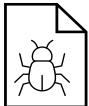
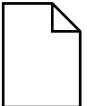






code			
detected			
correct?			



```
sum = 0
for i in range(10):
    sum += a[i] + b[i]
```

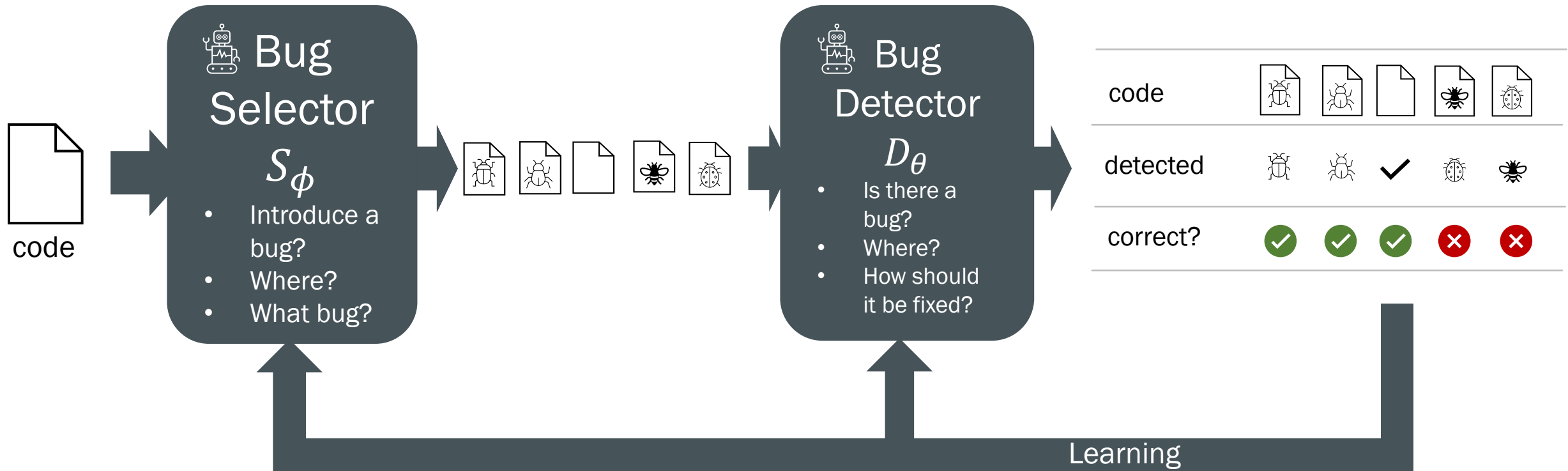
```
sum = 0
for i in range(10):
    sum += a[i] + b[email_address]
```

code			
detected			
correct?			

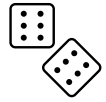


LEARNING



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

EVALUATION DATASETS



RandomBugs

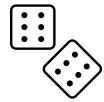
- ~700k random bugs
- Relatively Large
- Potentially non-representative of real bugs



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



By Francis Barlow - <http://mythfolklore.net/aesopica/barlow/59.htm>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=14476836>

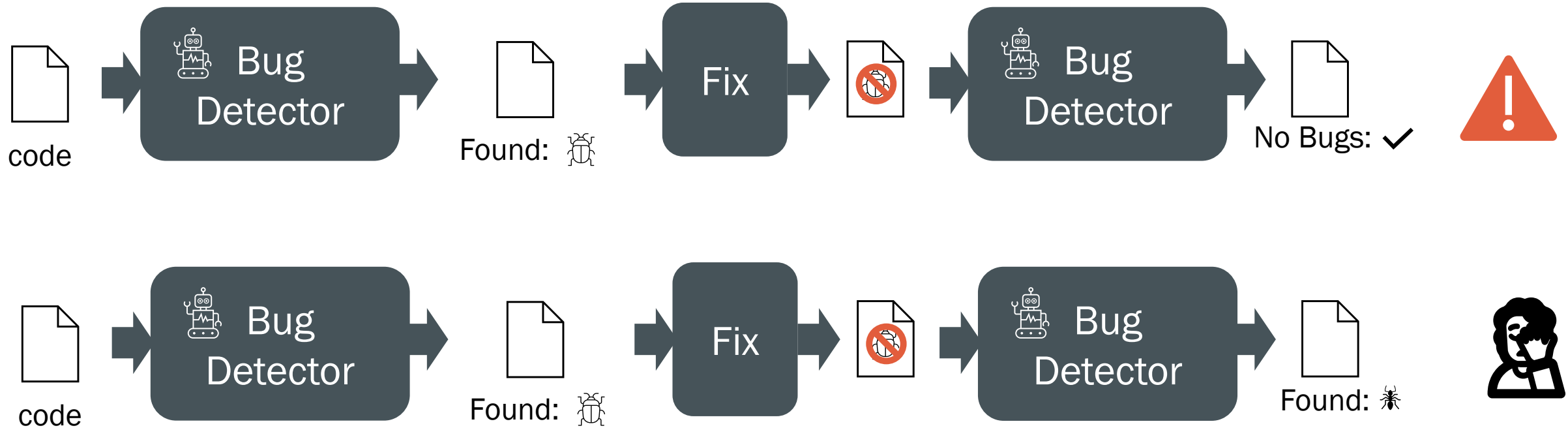
*program
analysis*

The ~~boy~~ who

error

cried ~~wolf~~

FILTERING THROUGH MODEL SELF-CONSISTENCY



FILTERING THROUGH MODEL SELF-CONSISTENCY



PyPIBugs+Fixed

	False Positive Rate	Accuracy
Before	88.1	48.9
After	73.5	46.1

85.8% of the filtering decisions were correct (filtered false positive)

```

@@ -213,27 +213,27 @@
213     213         )
214     214
215     215         # Return a room meeting info object created from the response JSON data
216     216         return self._object_factory("room_meeting_info", json_data)
217     217
218     218     def update(self, roomId, title, **request_parameters):
219     219         """Update details for a room, by ID.
220     220
221     221         Args:
222     222             roomId(basestring): The room ID.
223     223             title(basestring): A user-friendly name for the room.
224     224             **request_parameters: Additional request parameters (provides
225     225                 support for parameters that may be added in the future).
226     226
227     227         Returns:
228     228             Room: A Room object with the updated Webex Teams room details.
229     229
230     230         Raises:
231     231             TypeError: If the parameter types are incorrect.
232     232             ApiError: If the Webex Teams cloud returns an error.
233     233
234     234         """
235     235         check_type(roomId, basestring)
236     236         - check_type(roomId, basestring)
237     237         + check_type(title, basestring)
238     238
239     239         put_data = dict_from_items_with_values(
240     240             request_parameters,

```

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

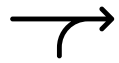
69 69 if include_meta_properties is not None:

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

↓



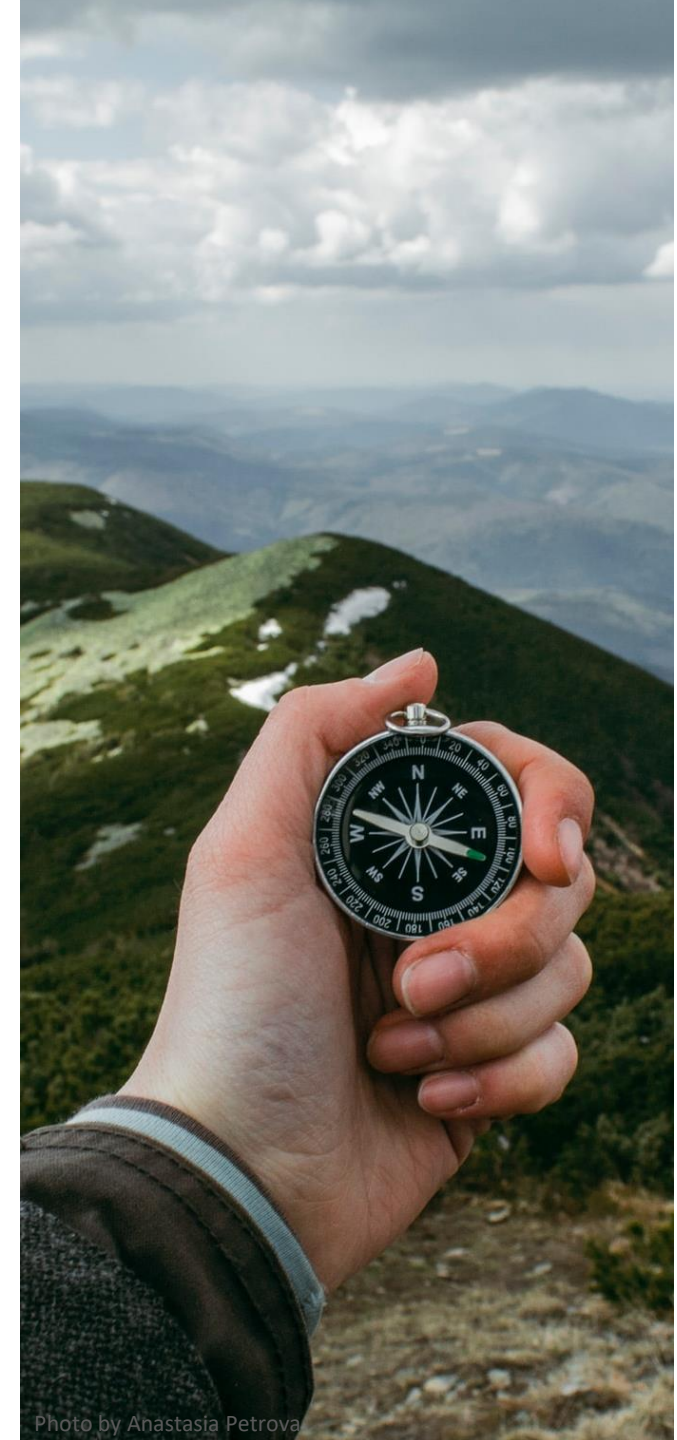
ML Models & AI Capabilities



Better fusion of PL and ML



Tools and UX



THE ML4CODE LANDSCAPE

Code Generation

Program Analysis

...

Code Completion

Program Synthesis

Specification Tuning

Specification Inference

Black-Box Analysis Learning

<https://ml4code.github.io>

CODE REPRESENTATION

Entities (Nodes)

- Tokens
- Non-Terminal Nodes
- Symbols

Relationships (Edges)

- Syntax
- AST Child
 - AST Sibling
 - Next Token

- Function Calls
- CandidateFormalArg
 - CandidateDocStringOf

Data Flow

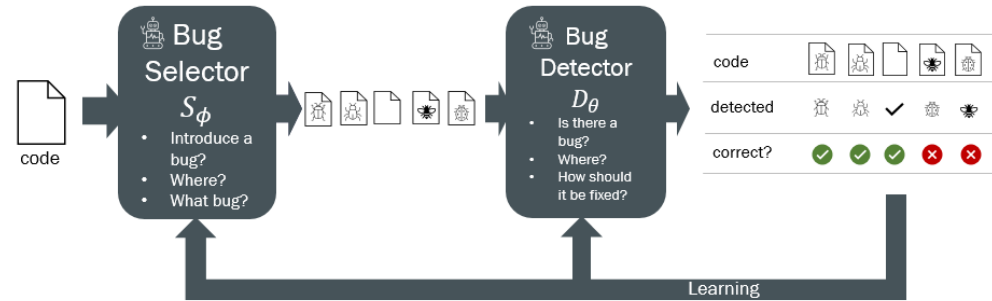
- MayFinalUseOf
- LastMayWrite
- NextMayUse

- Control Flow
- ControlFlowNext
 - AssignedFrom
 - ReturnsFrom
 - YieldsFrom

Symbols

- CandidateType
- OccurrenceOf
- CandidateMethodName

LEARNING



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

Bug Lab

REPAIRING CODE WITH MACHINE LEARNING

MILTOS ALLAMANIS



miltos.allamanis.com



@miltos1