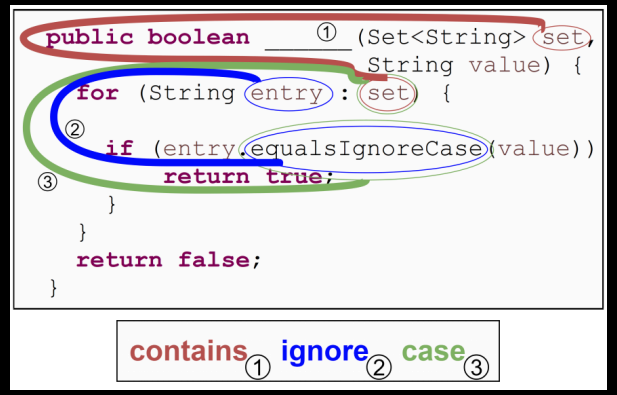
Neural Reverse Engineering of Stripped Binaries using Augmented Control Flow Graphs

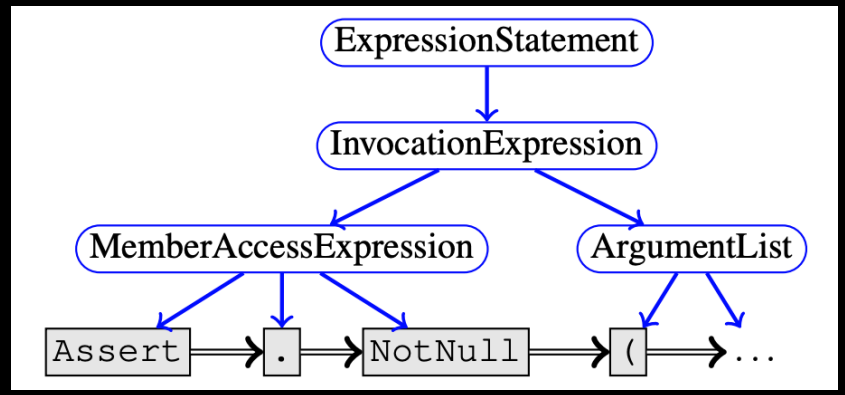


Yaniv David, Uri Alon, Eran Yahav
Technion, Israel

Great Progress Using ML on Source Code



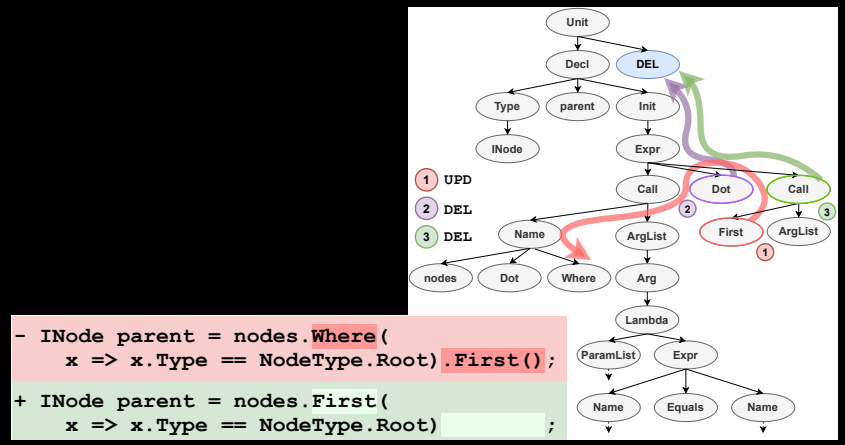
["code2seq", Alon et al., ICLR'2019]



[Program graphs, Allamanis et al., ICLR'2018]

```
void updateView() {
  // ...
  View v = ViewUtil.findView(name);
  if (v == null) {
    return;
  }
  data.send(((UpdatableView) v)
      .getContentMgt(), "UPDATE");
  // ...
}
```

["Getafix", Bader et al., OOPSLA'2019]



[Edit Completion, Brody et al., OOPSLA'2020]

Not a Lot of Progress for Binaries

- Why?

Source code in a high-level language

- Programmers follow common **patterns** (sort, merge, …)
- These patterns employ sequences of the language's **syntactic structures**
- Many **global** anchors: types, class structures, frameworks

```
void f(int[] a) {
    boolean s = true;
    for (int i = 0; i < a.length && s; i++) {
        s = false;
        for (int j = 0; j < a.length - 1 - i; j++) {
            if (a[j] > a[j+1]) {
                int temp = a[j];
                a[j] = a[j+1];
                a[j+1]= temp;
                s = true;
            }
        }
    }
}
```

Rich, structured, expressive & filled with hints

Disassembled Binary Code

```
...
mov rsi, rdi
mov rdx, 16
mov rax, [rbp-58h]
mov rdi, rax
mov edx, eax
mov rax, [rbp-4h]
mov rax, [rbp-58h]
mov rdi, rax
mov r8, 4
mov rdx, 10
mov esi, 0
lea rcx, [rbp-88h]
mov rdi, rax
...
```

- Machine generated, adhering to hardware specifications & limitations
- Few simple control flow instructions:
  - jump & call (+interrupt)
- Optimization causes: entangled computation flows, context dependent

Long unstructured seqeunce of low-level operations

Disassembled Binary Code

```
...
mov rsi, rdi
mov rdx, 16
mov rax, [rbp-58h]
mov rdi, rax
mov edx, eax
mov rax, [rbp-4h]
mov rax, [rbp-58h]
mov rdi, rax
mov r8, 4
mov rdx, 10
mov esi, 0
lea rcx, [rbp-88h]
mov rdi, rax
...
```

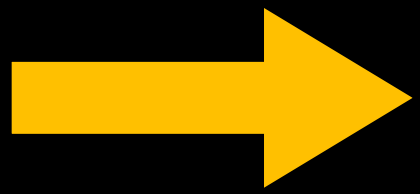
- Machine generated, adhering to hardware specifications & limitations
- Optimiz… uses: entangled computat… ows, context dependent

These put a lot of pressure on the model

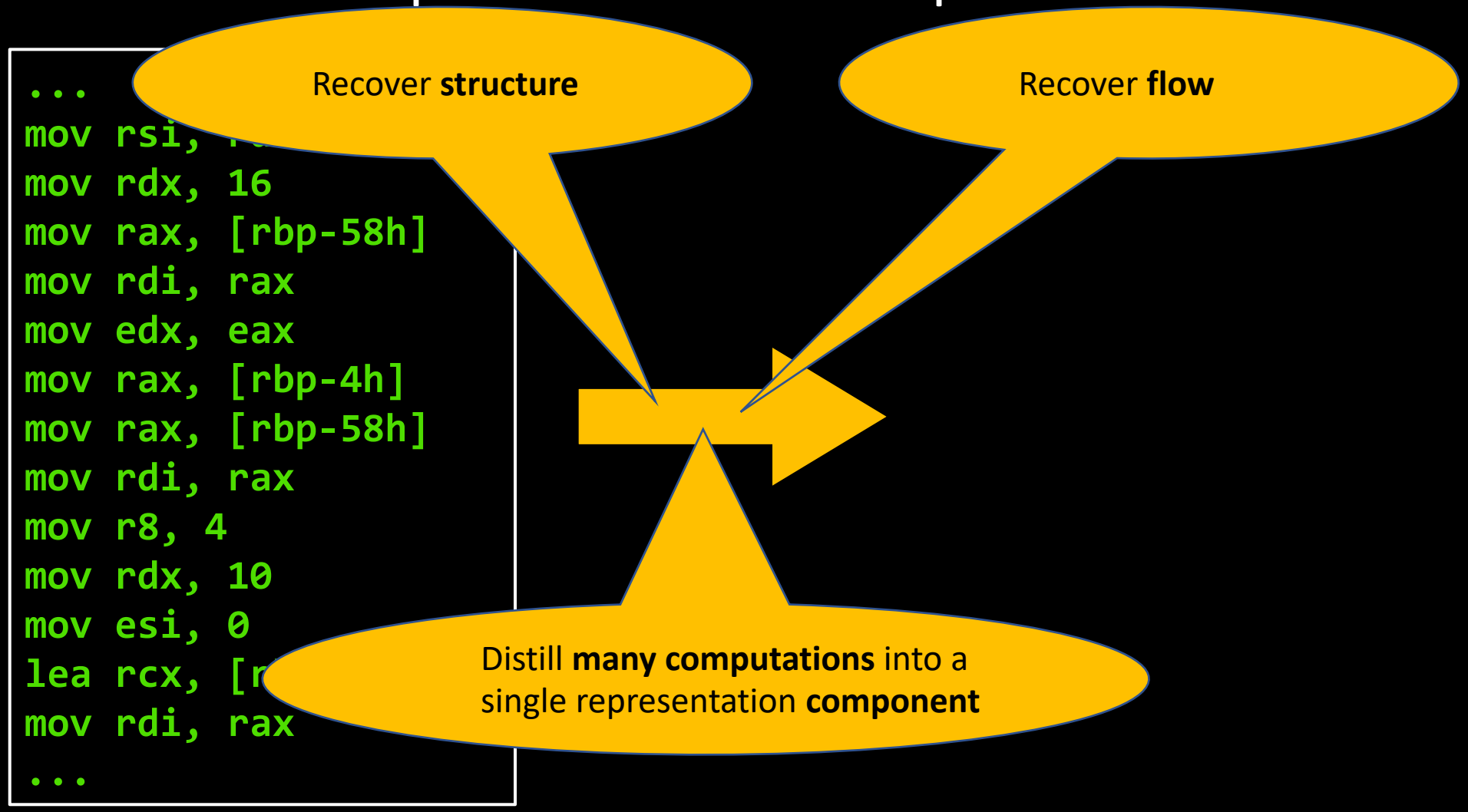
Long unstructured seqeunce of low-level operations

Key Idea: Use Binary Program Analysis to Create a Compact & Rich Representation

```
...
mov rsi, rdi
mov rdx, 16
mov rax, [rbp-58h]
mov rdi, rax
mov edx, eax
mov rax, [rbp-4h]
mov rax, [rbp-58h]
mov rdi, rax
mov r8, 4
mov rdx, 10
mov esi, 0
lea rcx, [rbp-88h]
mov rdi, rax
...
```

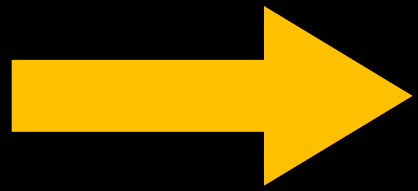


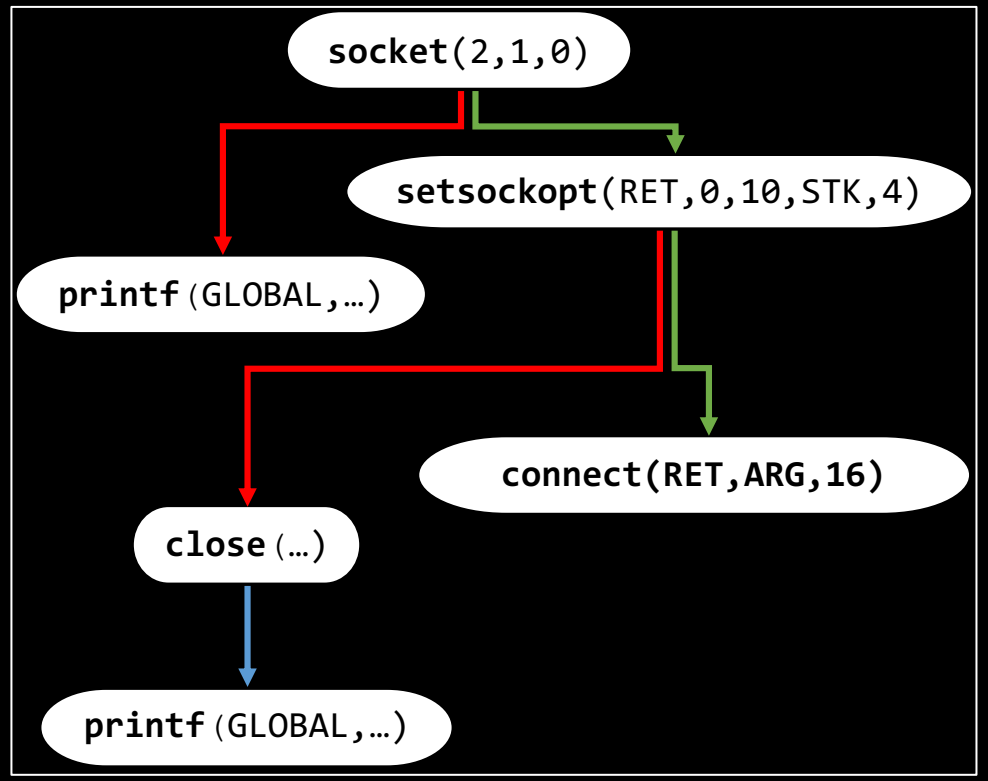
Key Idea: Use Binary Program Analysis to Create a Compact & Rich Representation



Key Idea: Use Binary Program Analysis to Create a Compact & Rich Representation

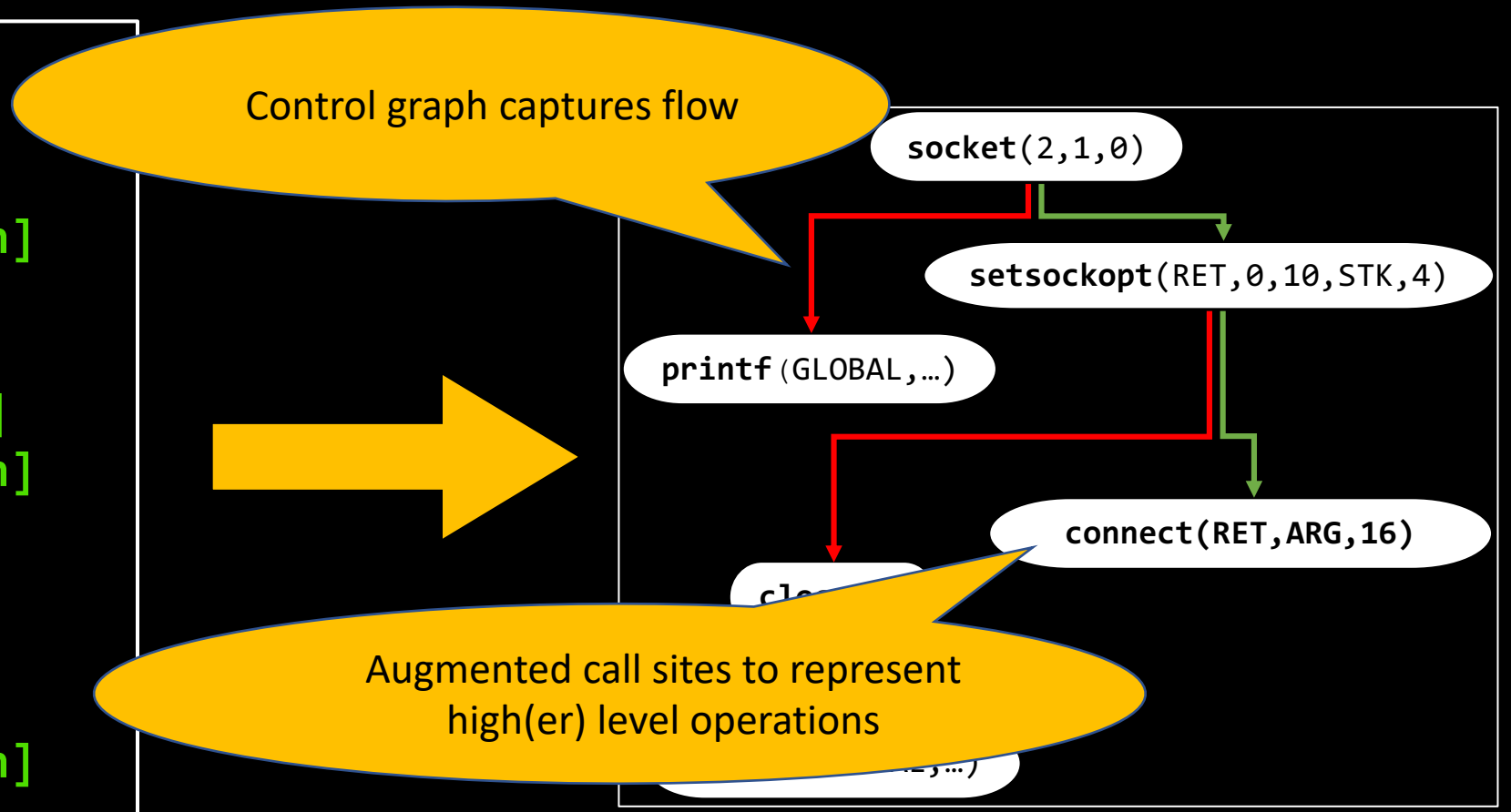
```
...
mov rsi, rdi
mov rdx, 16
mov rax, [rbp-58h]
mov rdi, rax
mov edx, eax
mov rax, [rbp-4h]
mov rax, [rbp-58h]
mov rdi, rax
mov r8, 4
mov rdx, 10
mov esi, 0
lea rcx, [rbp-88h]
mov rdi, rax
...
```





Key Idea: Use Binary Program Analysis to Create a Compact & Rich Representation

```
...
mov rsi, rdi
mov rdx, 16
mov rax, [rbp-58h]
mov rdi, rax
mov edx, eax
mov rax, [rbp-4h]
mov rax, [rbp-58h]
mov rdi, rax
mov r8, 4
mov rdx, 10
mov esi, 0
lea rcx, [rbp-88h]
mov rdi, rax
...
```



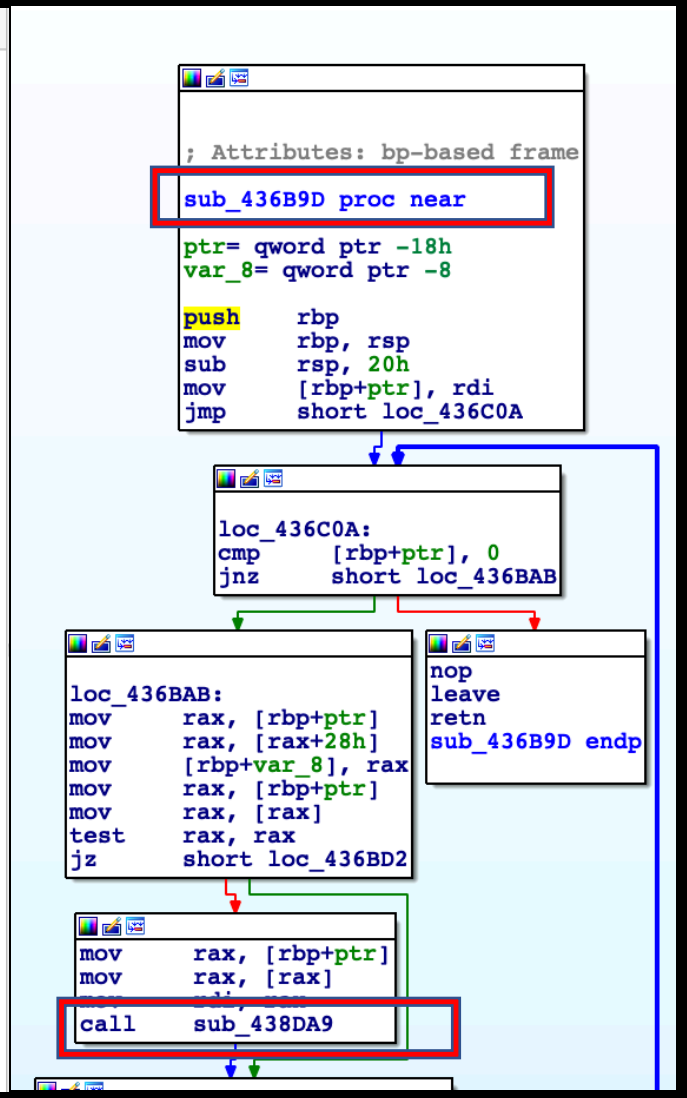
Motivating Task

Naming Procedures in Binaries

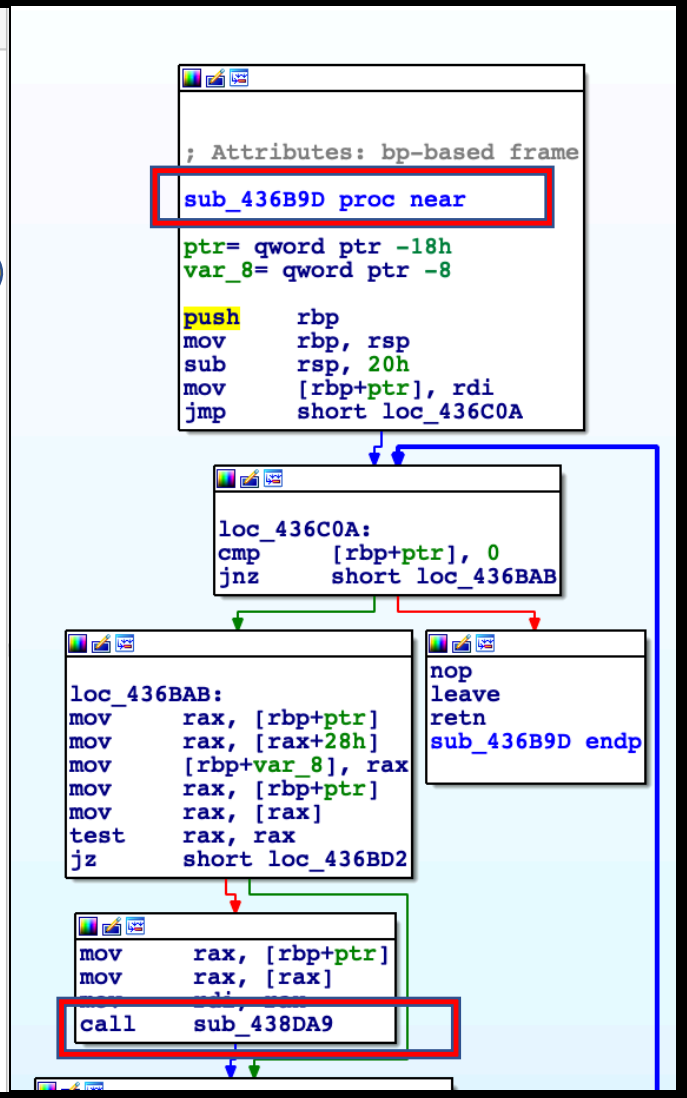
Helping Reverse Engineers

Function name

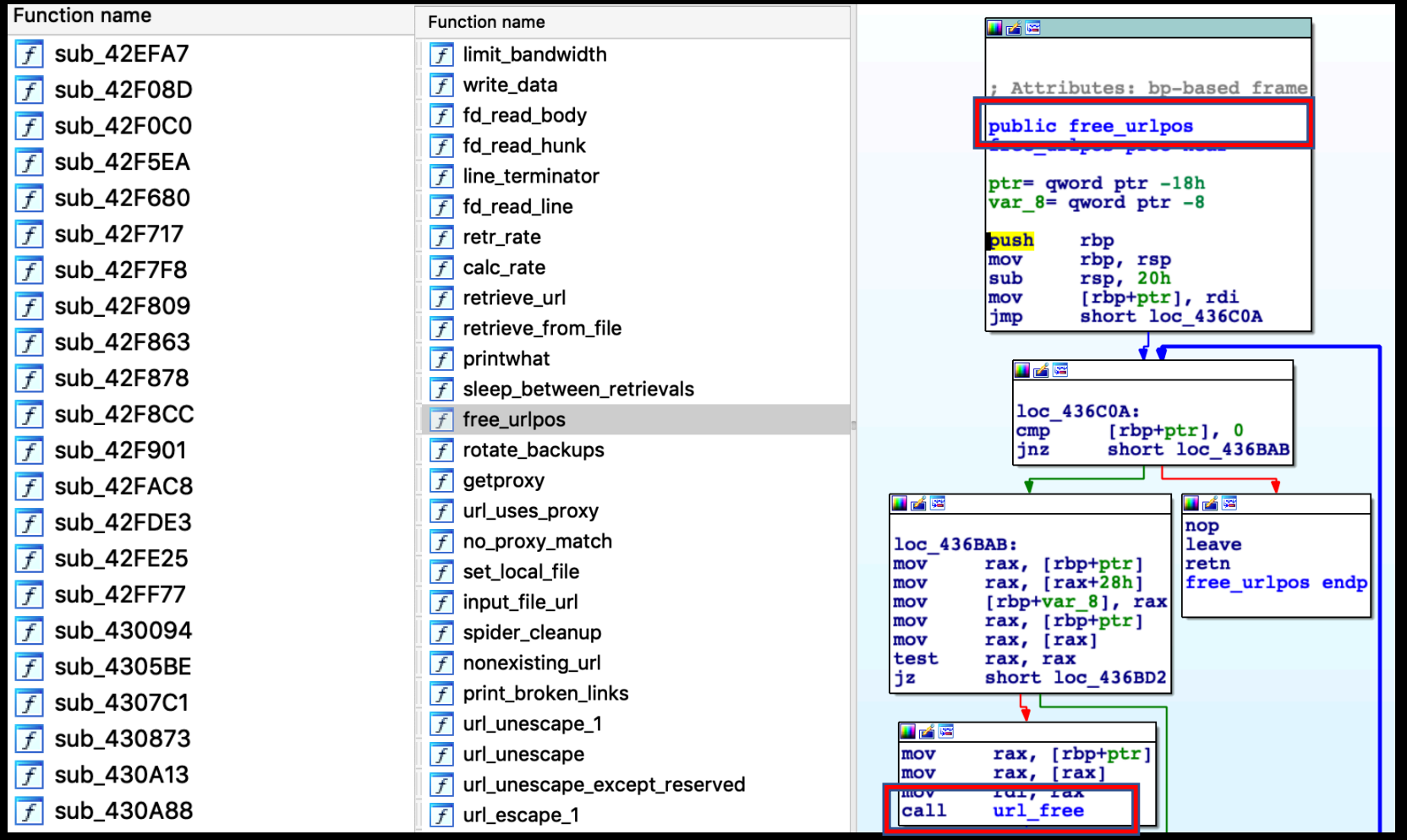
- sub_42EFA7
- sub_42F08D
- sub_42F0C0
- sub_42F5EA
- sub_42F680
- sub_42F717
- sub_42F7F8
- sub_42F809
- sub_42F863
- sub_42F878
- sub_42F8CC
- sub_42F901
- sub_42FAC8
- sub_42FDE3
- sub_42FE25
- sub_42FF77
- sub_430094
- sub_4305BE
- sub_4307C1
- sub_430873
- sub_430A13
- sub_430A88



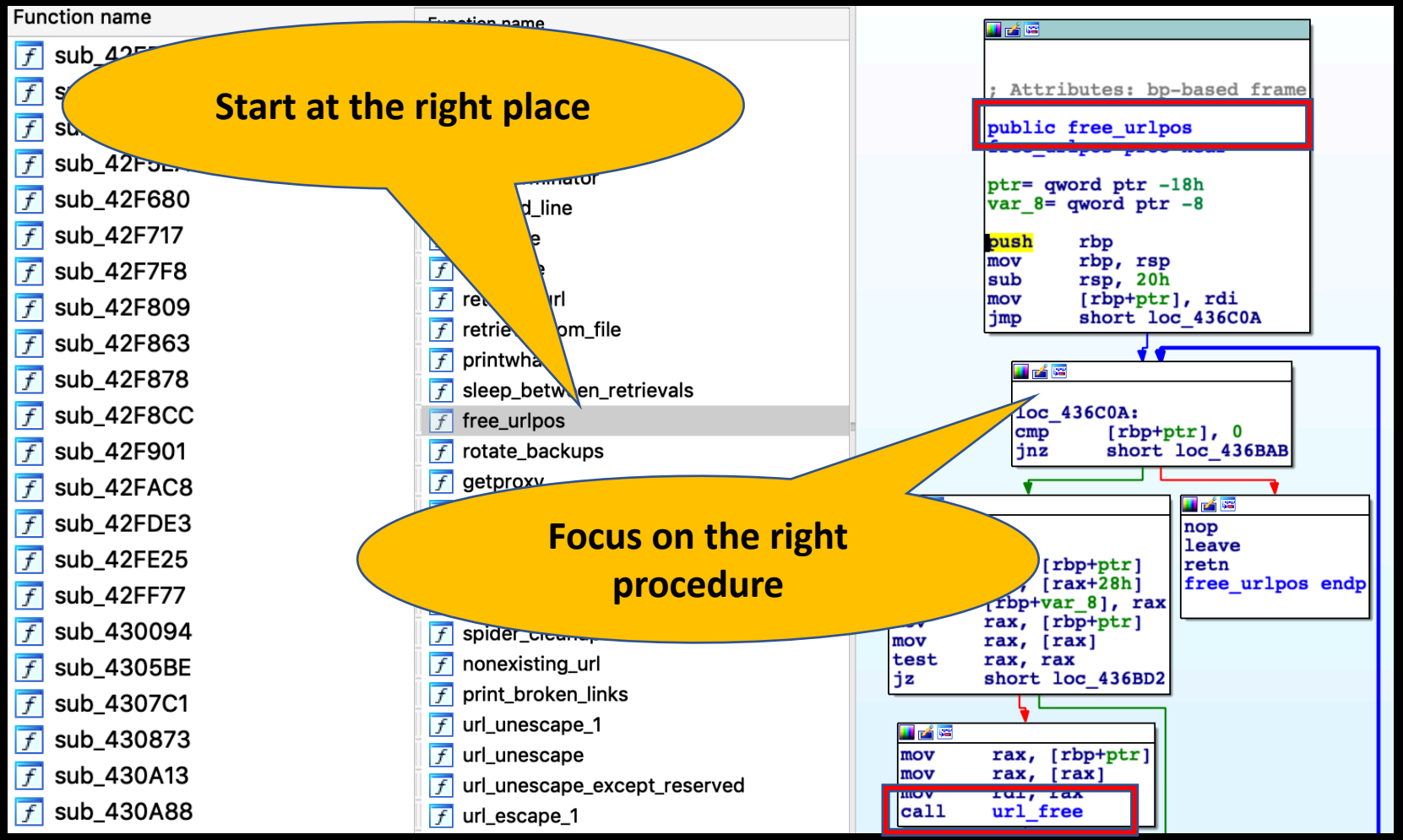
Helping Reverse Engineers



Un-Stripping Procedure Names

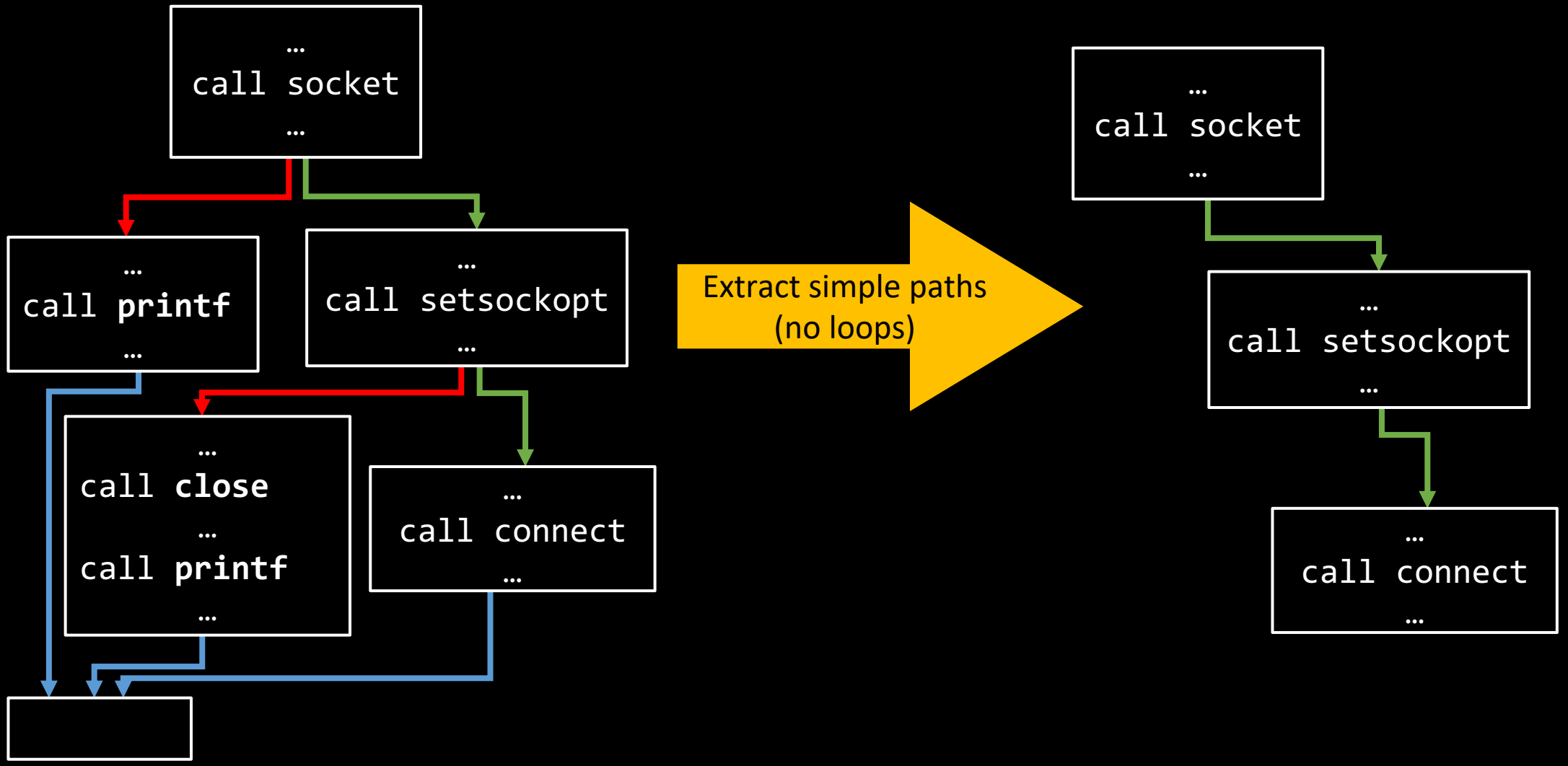


Un-Stripping Procedure Names

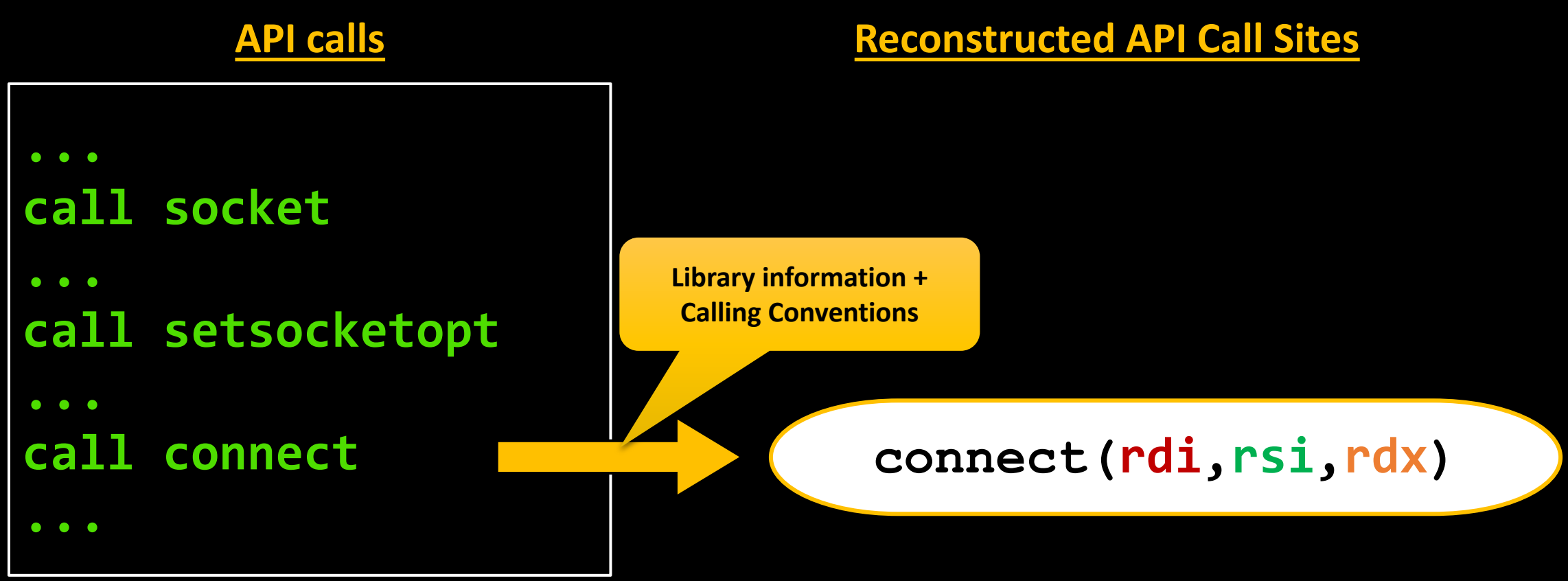


Our Approach

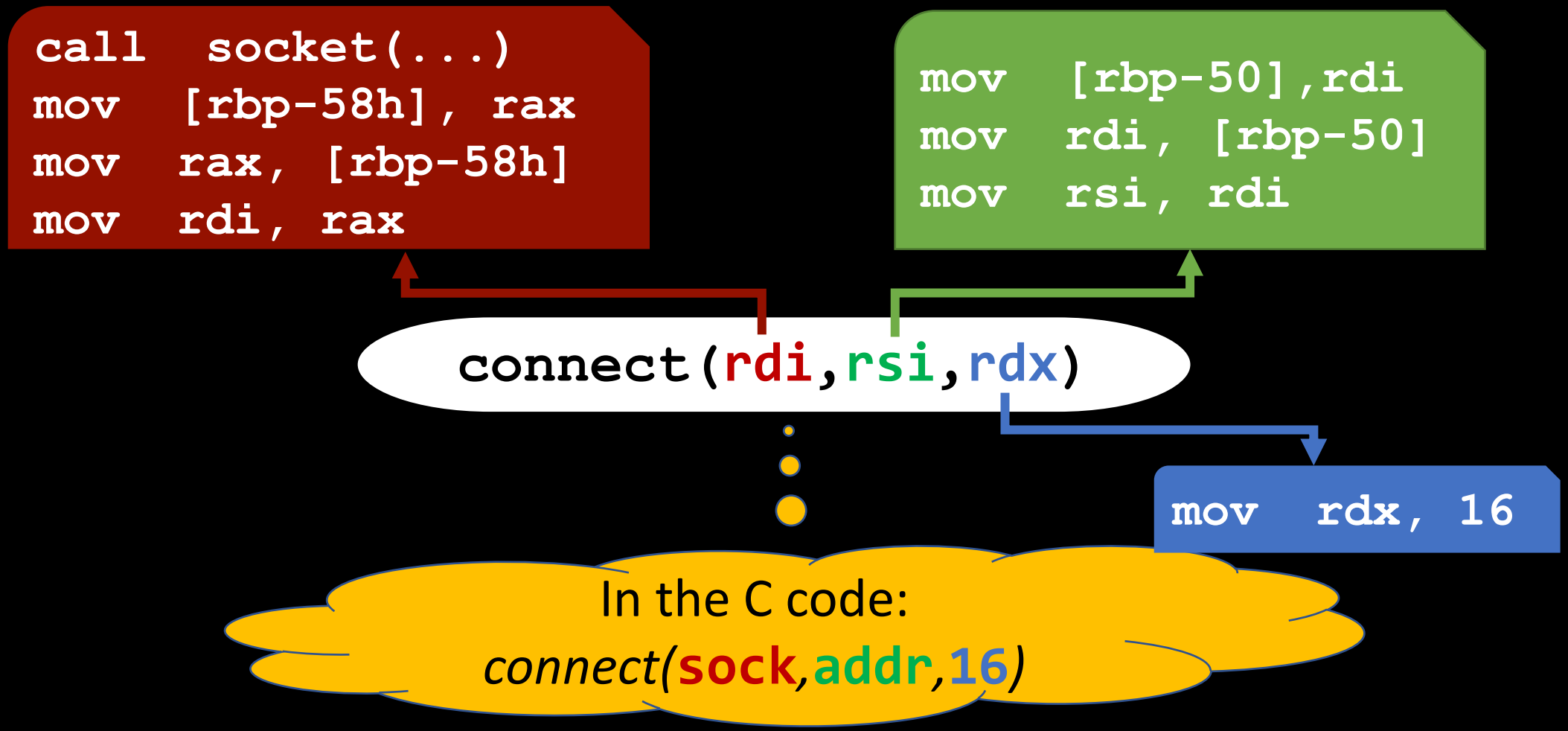
Extract Paths From the CFG



Using API Calls

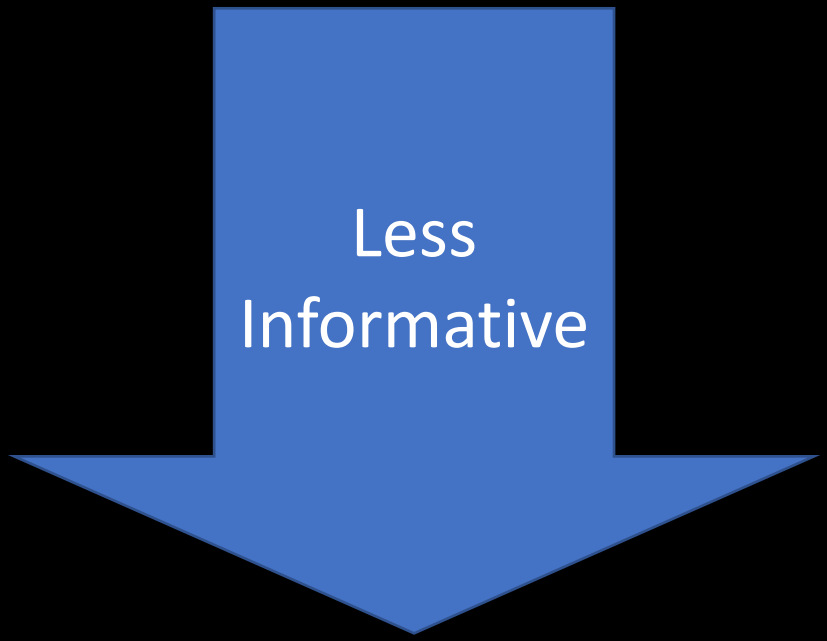


Augmenting Call Sites

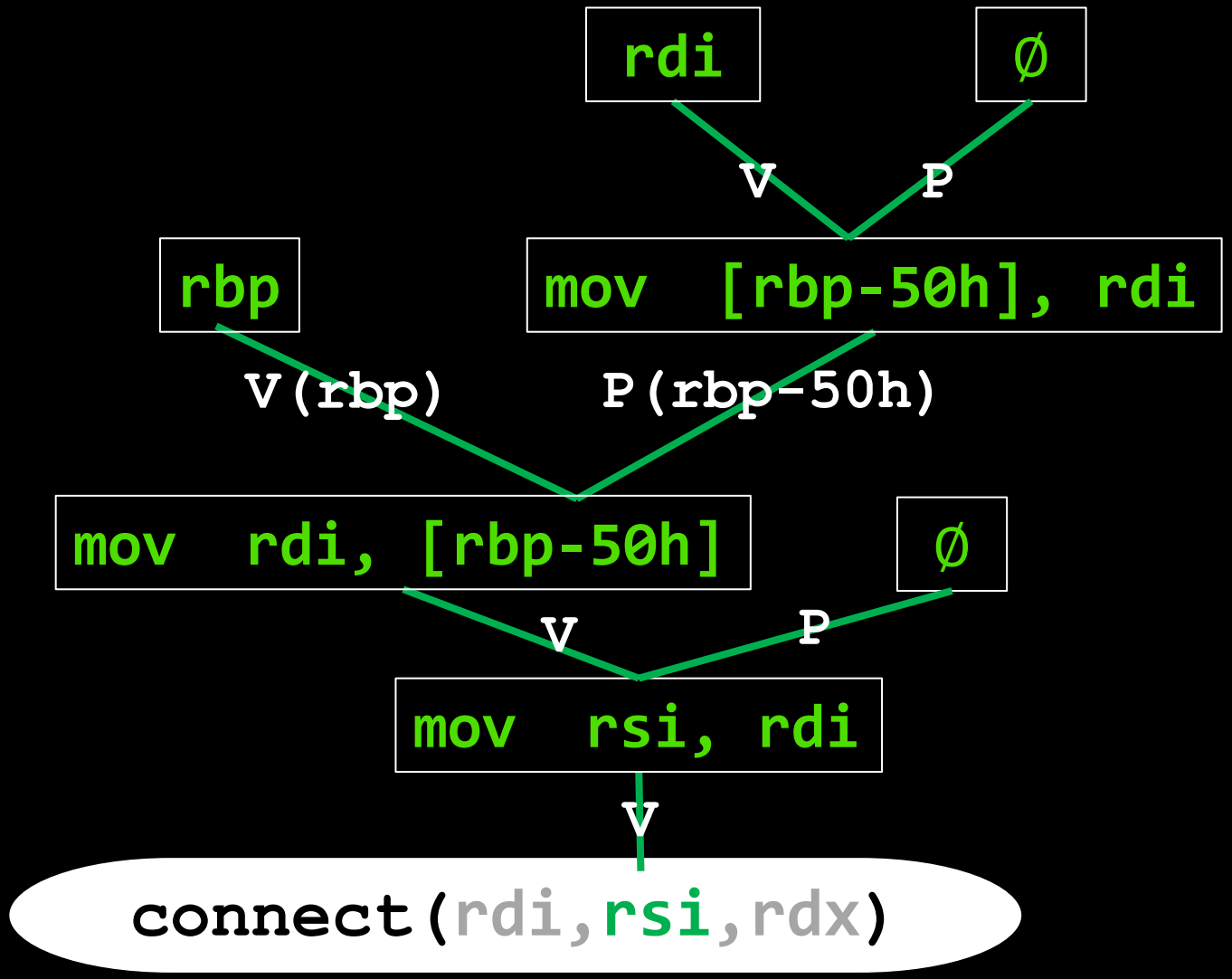


Using Concrete or Abstracted Values

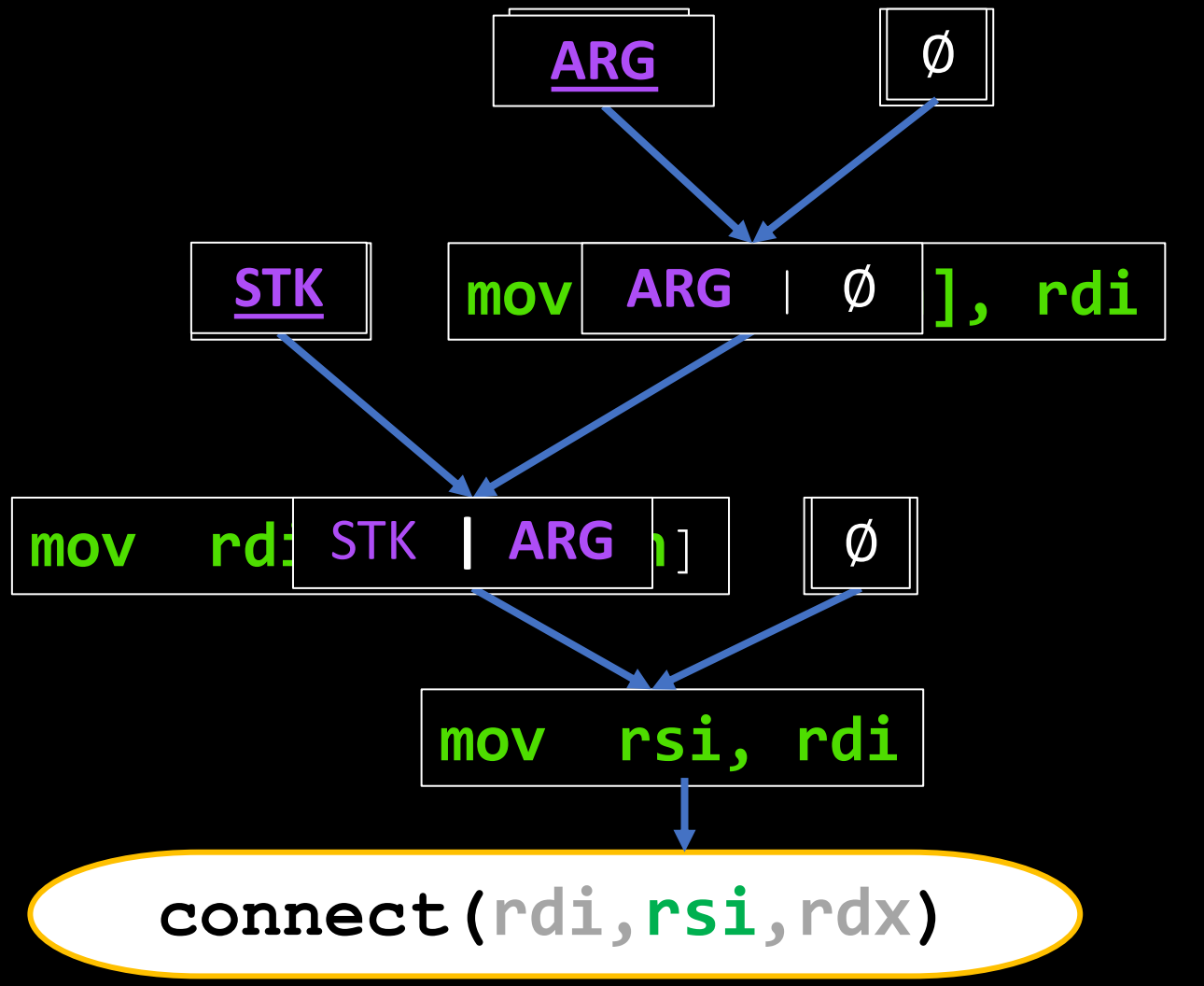
1. Concrete value (Integer, Enum, String)
2. ARG – procedure argument
3. GLOBAL - pointer to a global variable
4. RET – a return value from a call
5. STACK – pointer to stack memory
6. Ø – no information



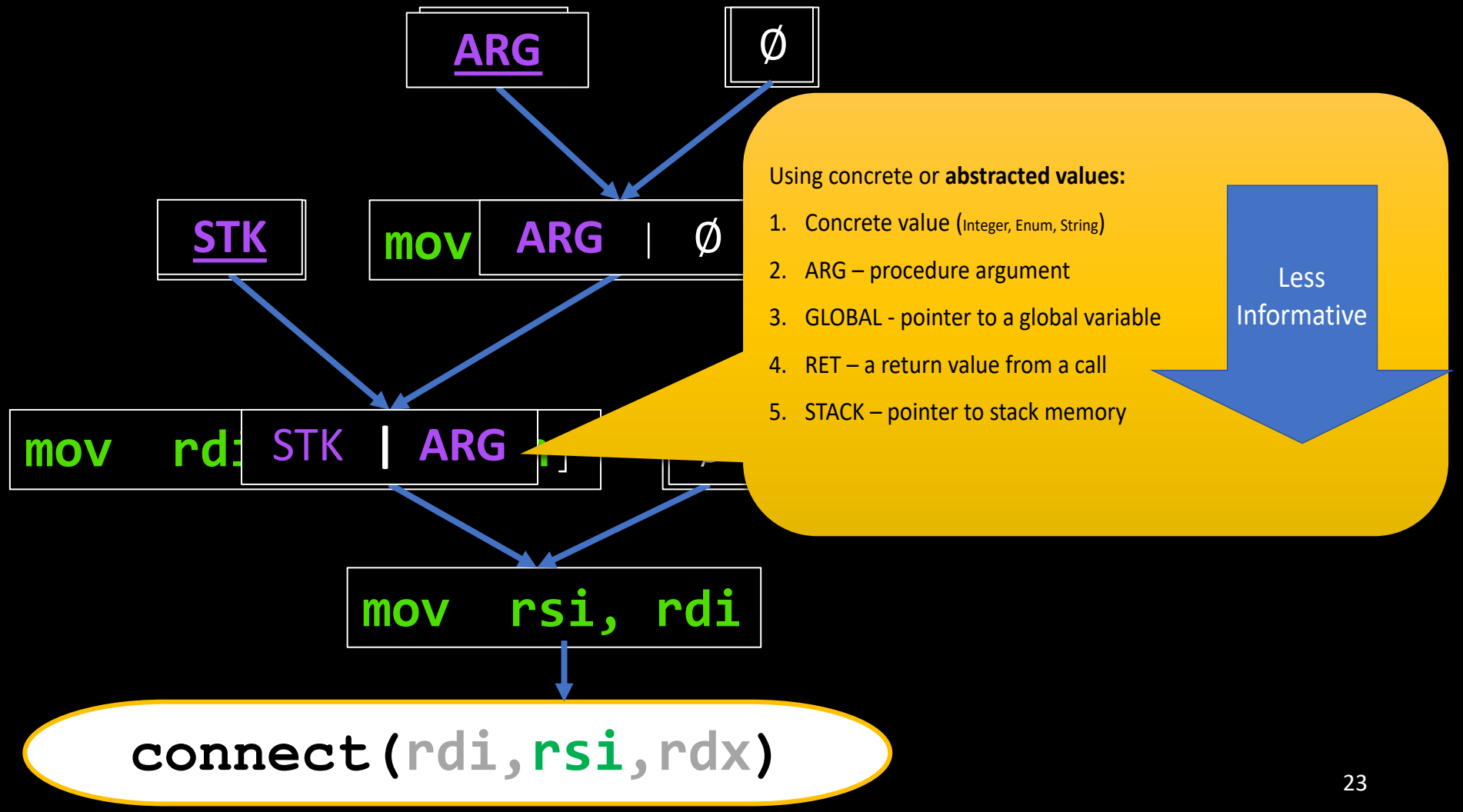
Pointer-Aware Slicing of Call Site Arguments



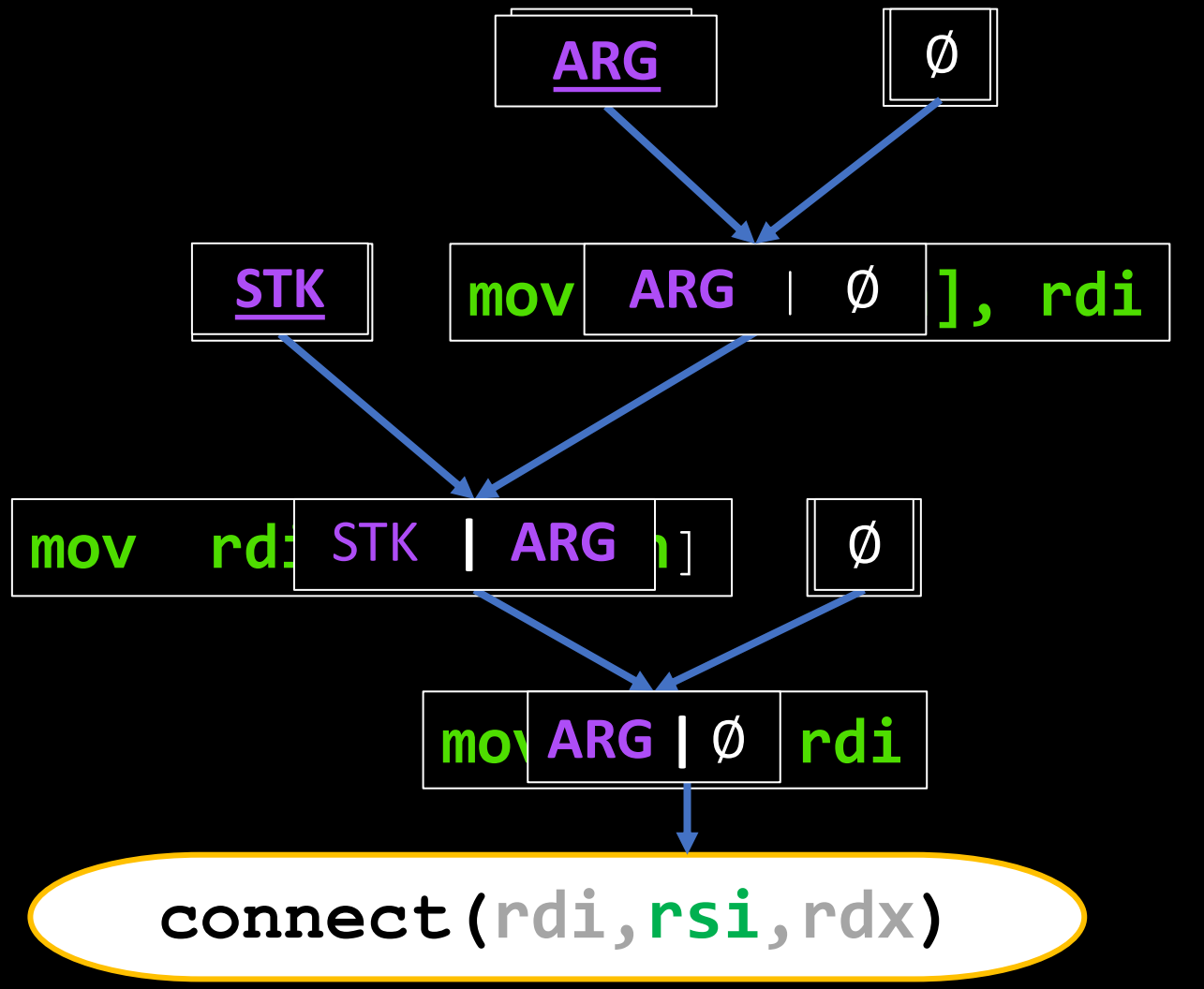
Augmenting Call Site Arguments



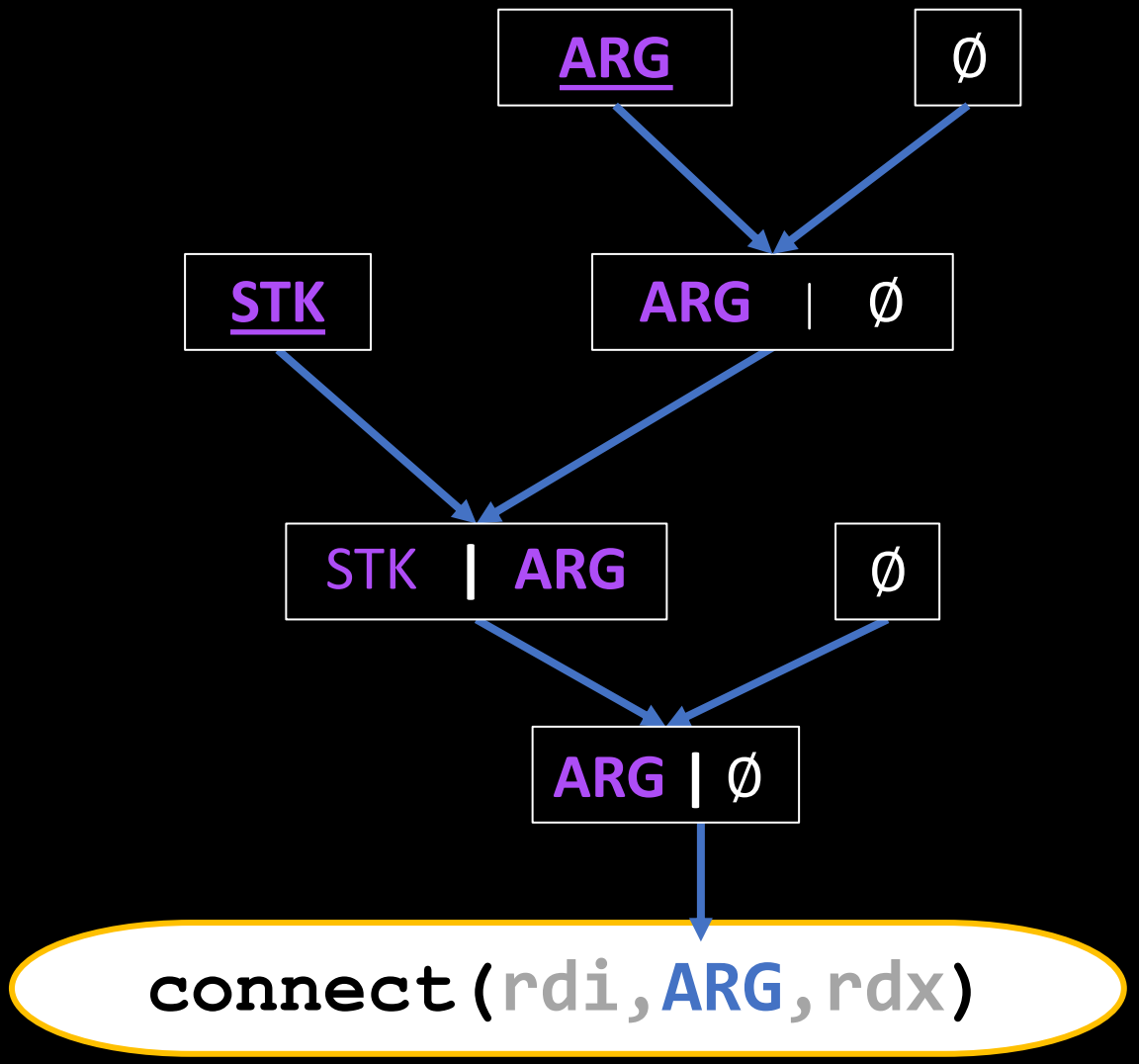
Augmenting Call Site Arguments



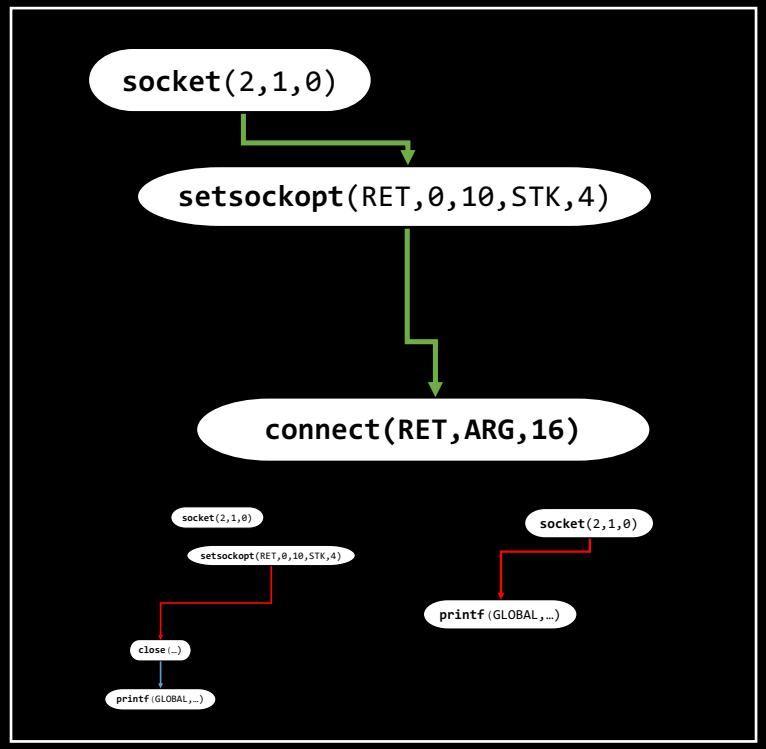
Augmenting Call Site Arguments



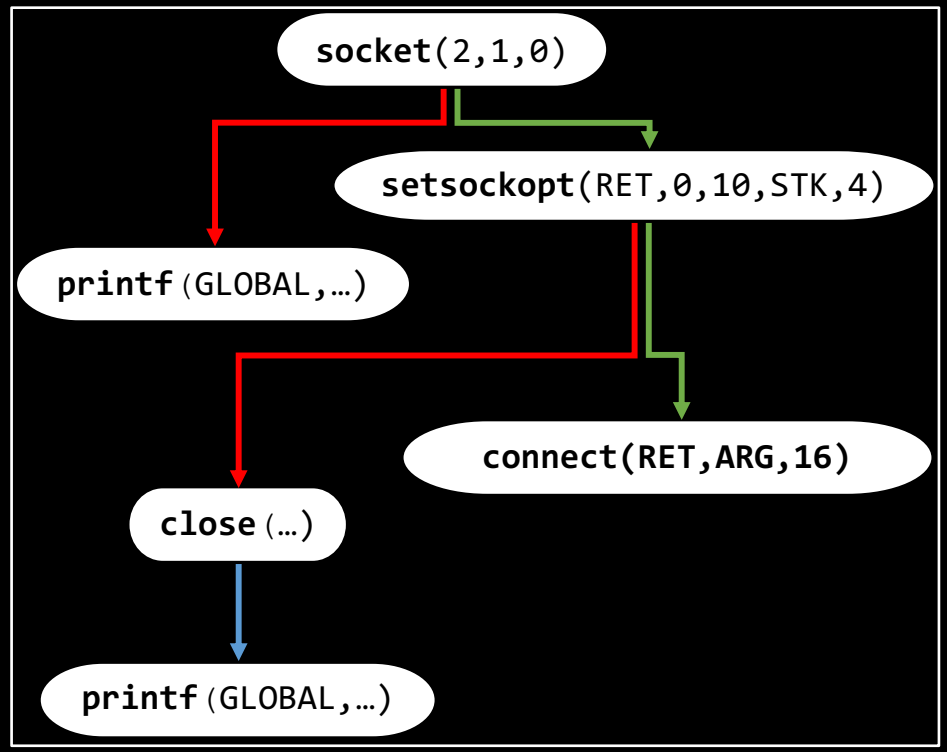
Augmenting Call Site Arguments



Augmented Control Flow Graph



Place augmented call sites back in the CFG



LSTM & Transformer

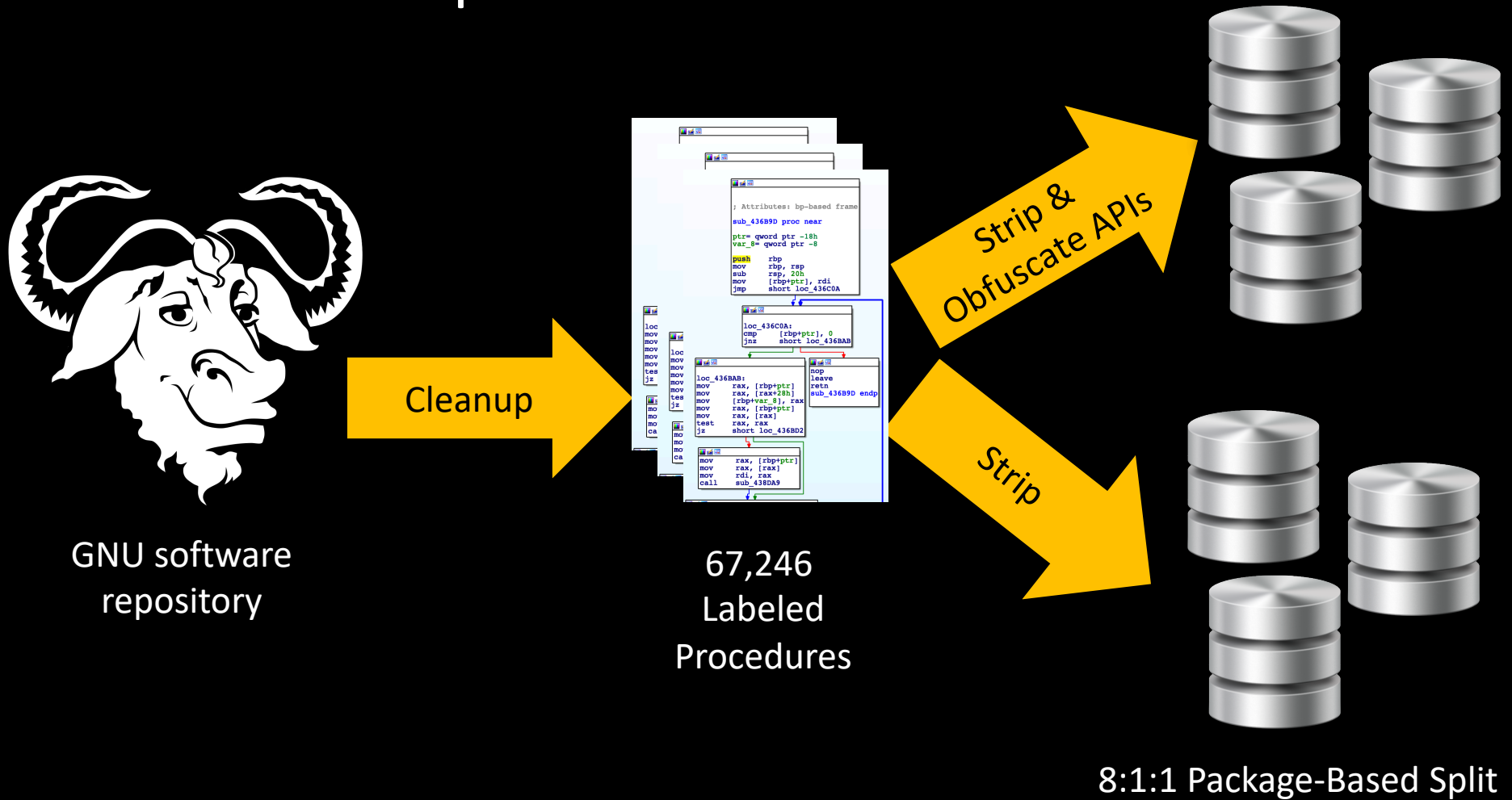
GNN

Evaluation

Implementation: Nero

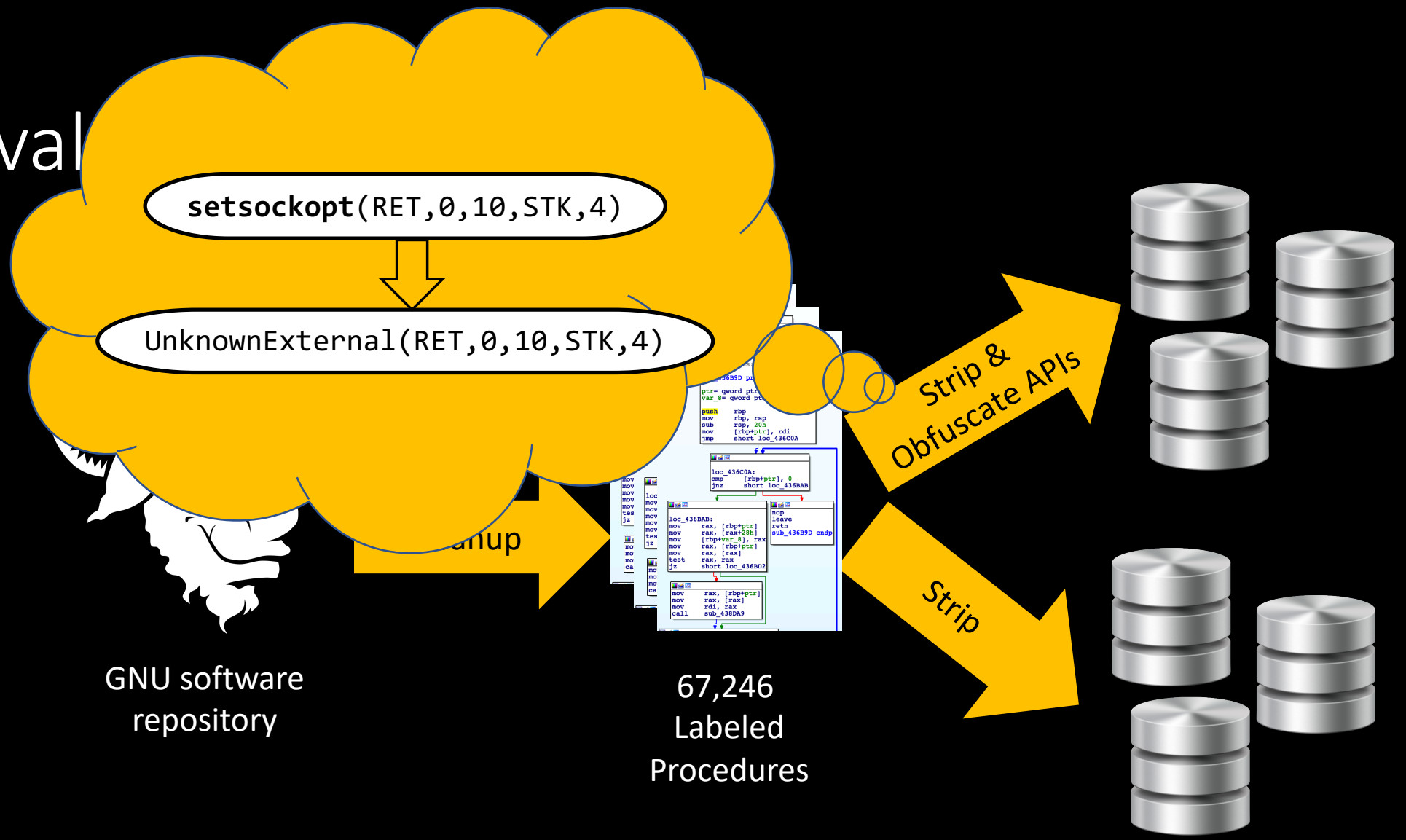
https://github.com/tech-srl/Nero

Evaluation Corpus



zenodo https://doi.org/10.5281/zenodo.4081641

Eval



GNU software repository

67,246 Labeled Procedures

8:1:1 Package-Based Split

zenodo https://doi.org/10.5281/zenodo.4081641

Evaluation Results

Nero-GNN

Nero-Transformer

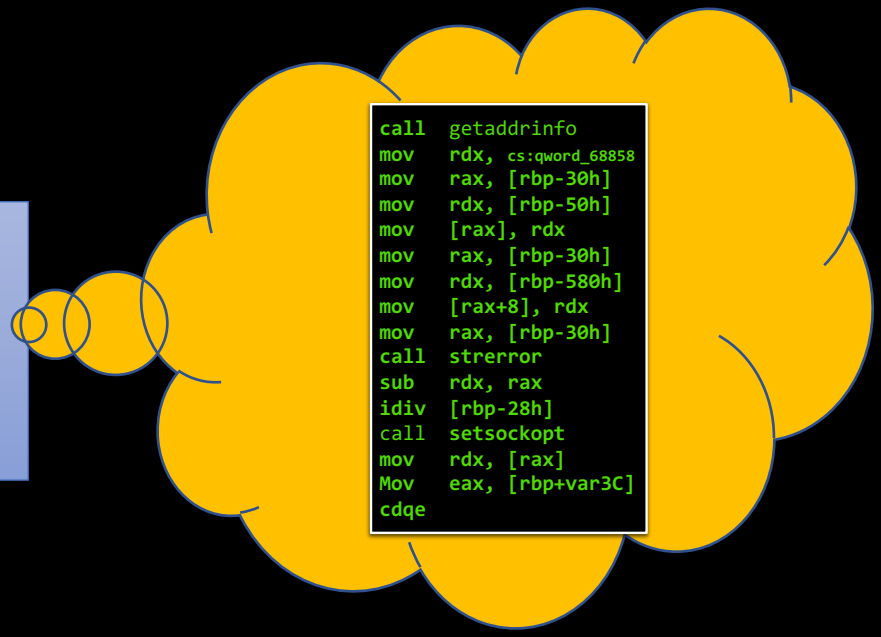
Nero-LSTM

DIRE [Lacomis et al. 2019]

Debin [He et al. 2018]

LSTM-text

Transformer-text



Evaluation Results

Nero-GNN

Nero-Transformer

Nero-LSTM

DIRE [Lacomis et al. 2019]

Debin [He et al. 2018]

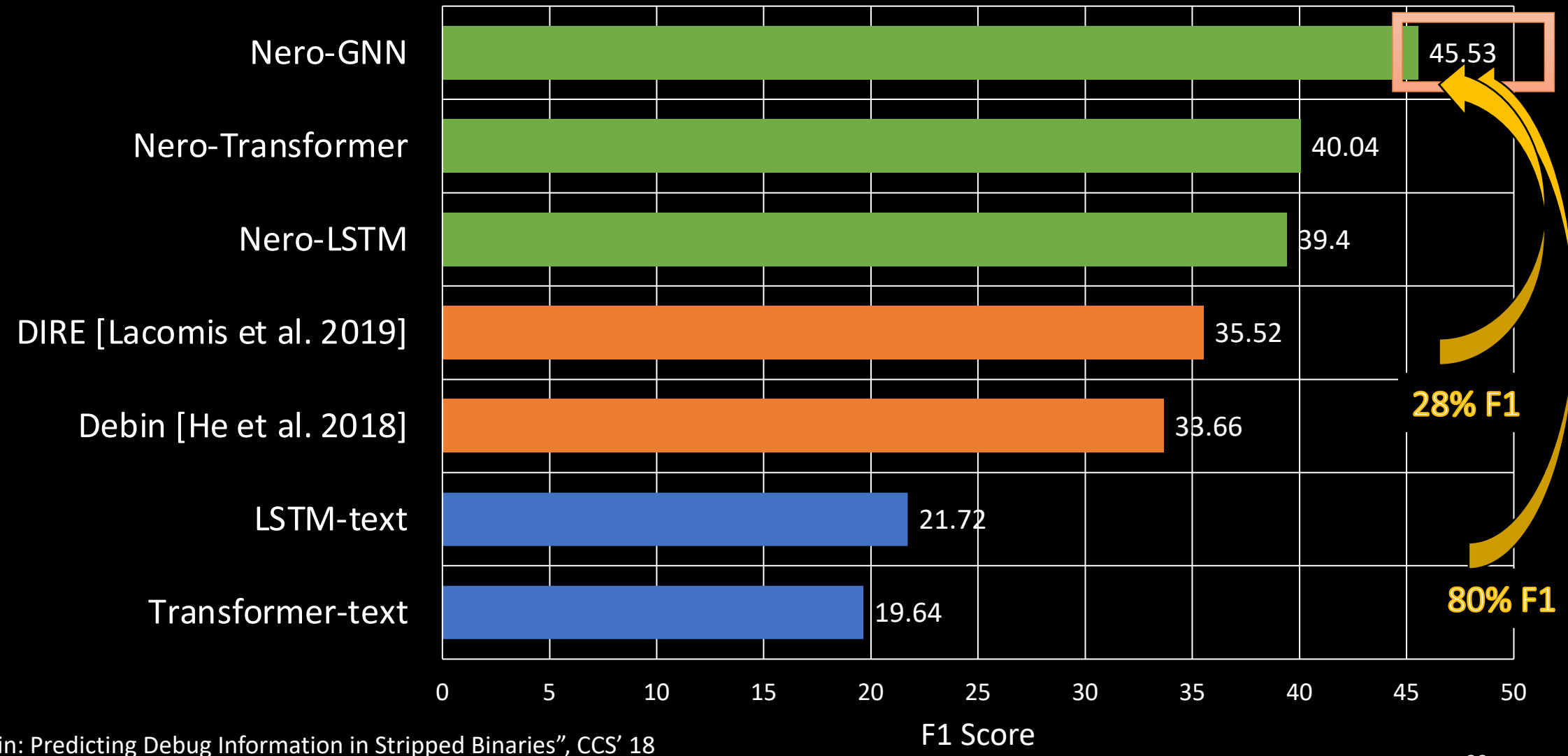
LSTM-text

Transformer-text

”Debin: Predicting Debug Information in Stripped Binaries”, CCS’ 18

”DIRE: A Neural Approach to Decompiled Identifier Naming”, ASE ’19

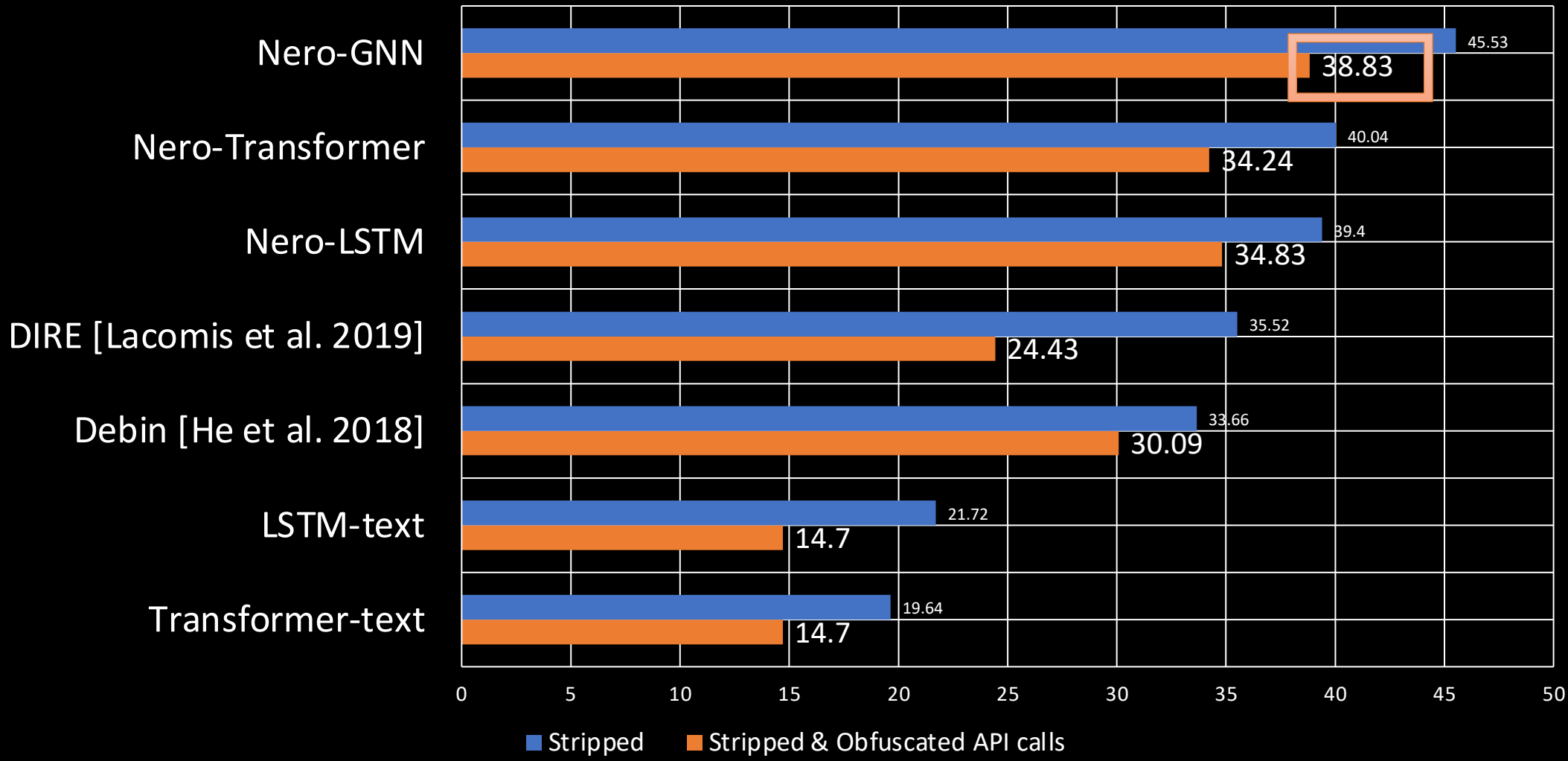
Evaluation Results



”Debin: Predicting Debug Information in Stripped Binaries”, CCS’ 18

”DIRE: A Neural Approach to Decompiled Identifier Naming”, ASE ’19

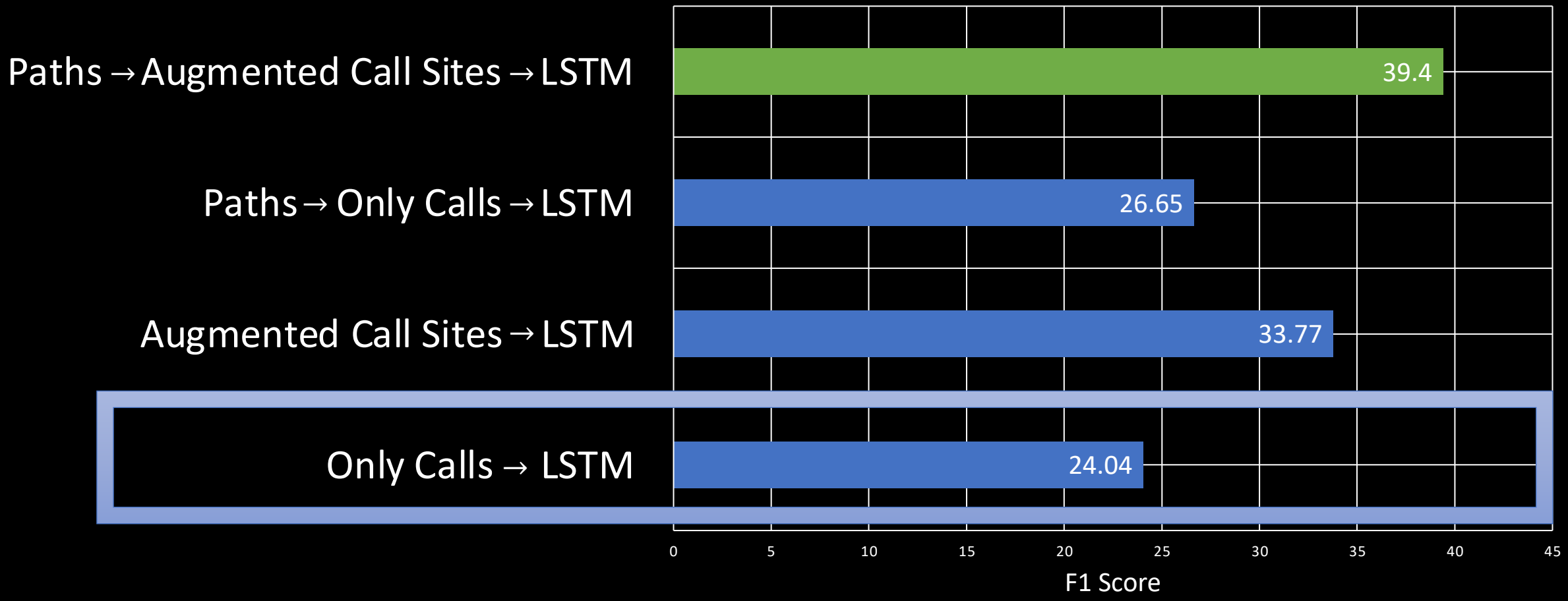
Evaluation Results



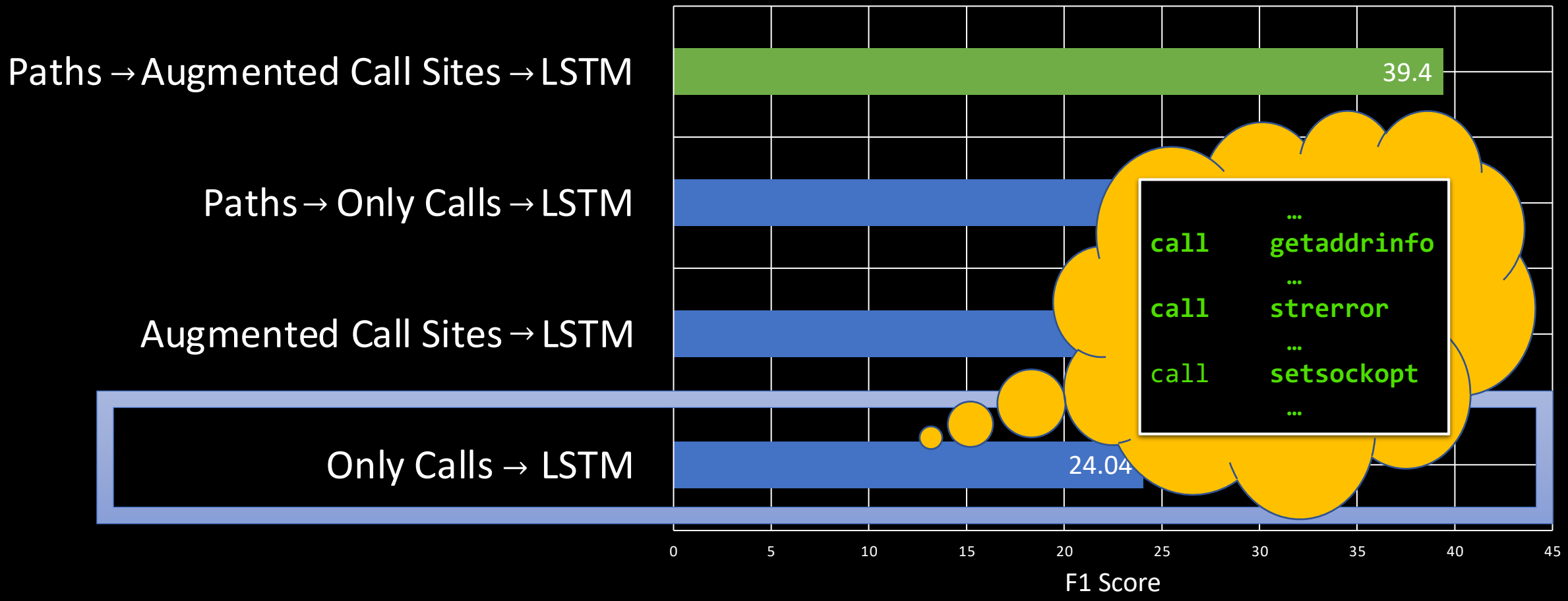
”Debin: Predicting Debug Information in Stripped Binaries”, CCS’ 18

”DIRE: A Neural Approach to Decompiled Identifier Naming”, ASE ’19

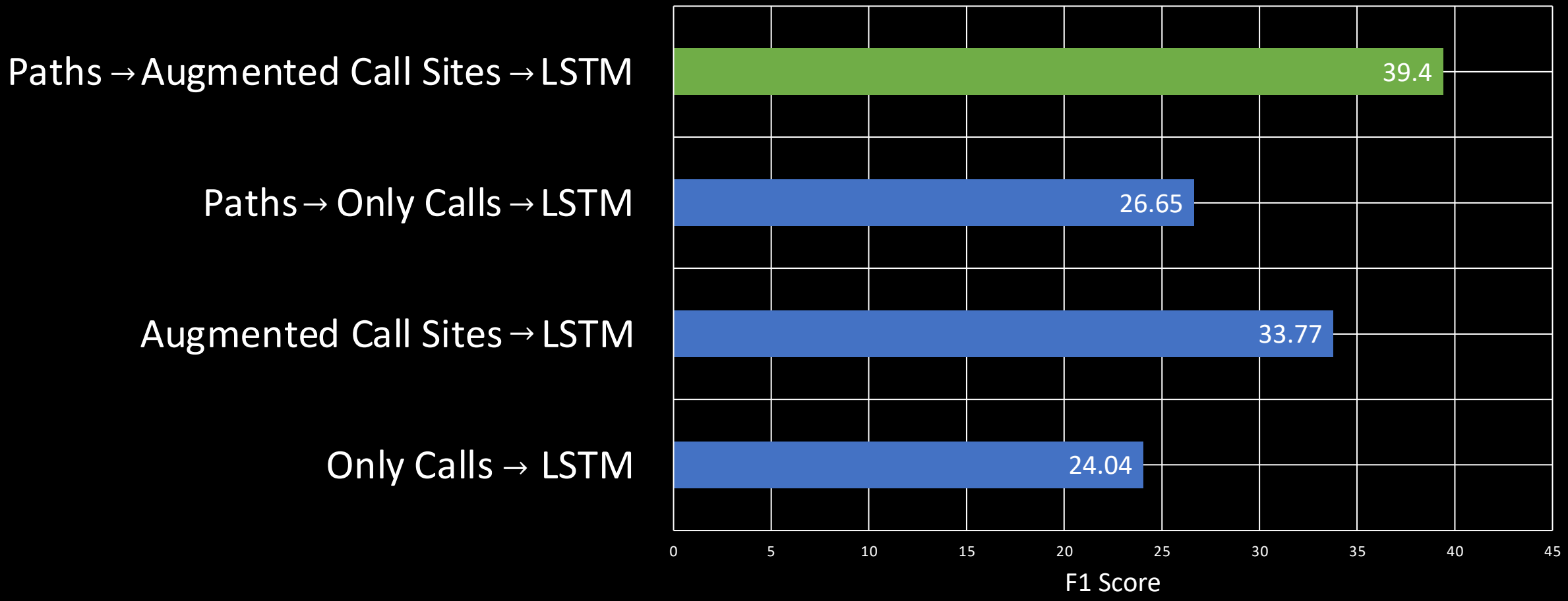
Ablation Study



Ablation Study



Ablation Study



Qualitive Evaluation

Error Type	Ground Truth	Predicted Name
Programmers Vs English Language	i18n_**initialize**	i18n_**init**
	split_**cfg**_path	split_**config**_path
	add_env_**opt**	add_**option**
Date Structure Name Missing	get_**best_speed**	get_**list_item**
	ftp_parse_**winnt_ls**	parse_**tree**
	abort_**gzip**_signal	fatal_signal_handler
Verb Replaced	**read**_units	**parse**
	retrieve_from_file	**get**_from_file
	display_help	**show**_help

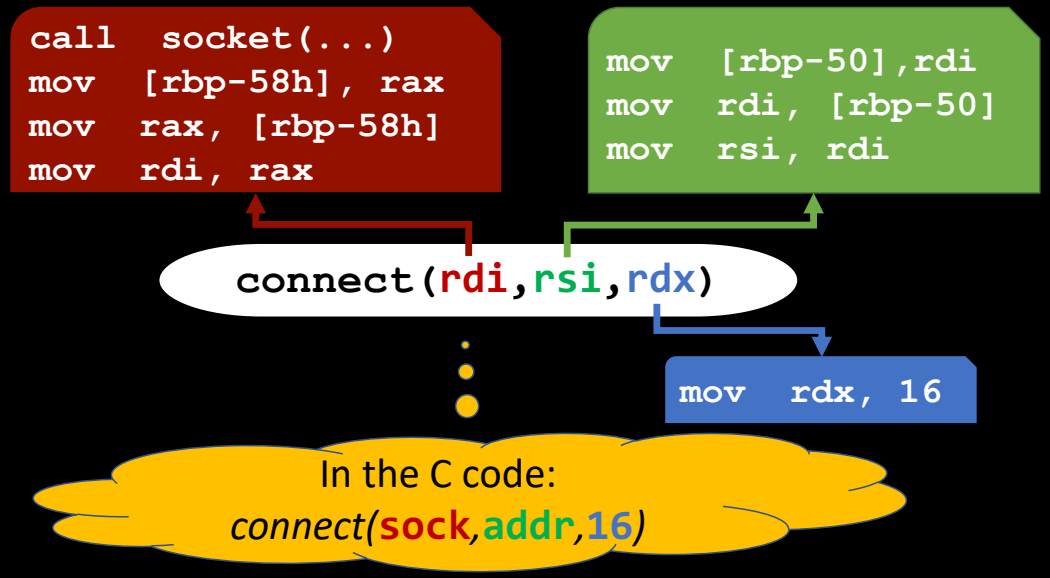
Qualitive Evaluation

Error Type	Ground Truth	Predicted Name
Programmers Vs English Language	i18n_**initialize**	i18n_**init**
	split_**cfg**_path	
	add_env_**opt**	
Date Structure Name Missing	get_**best_speed**	**get_item**
	ftp_parse_**winnt_ls**	parse_**tree**
	abort_**gzip**_signal	fatal_signal_handler
Verb Replaced	**read**_units	**parse**
	retrieve_from_file	**get**_from_file
	display_help	**show**_help

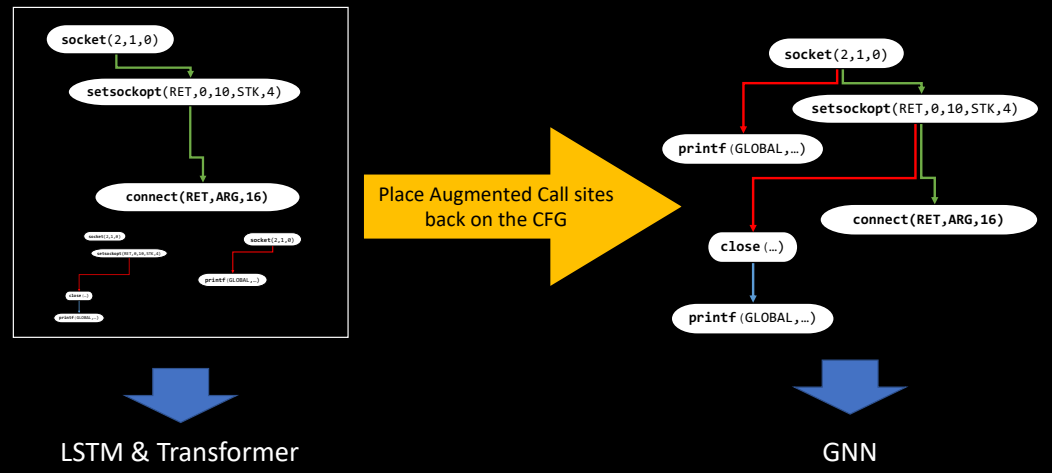
Measured F1 is actually a lower-bound

Takeaway Messages

Augmented call sites serve as a strong basis for procedure representations



Reconstructing the CFG enables the use of seq2seq and GNN models



https://github.com/tech-srl/Nero