SOFTWARE CLONES:
− Pieces of code that are similar (syntactically or semantically)...  
− May be a result of a cut and paste

THE PROBLEM:
− They can cause copyright issues; 
− They can cause inherited maintenance issues; 
− They can be difficult to find, given that evolve independently (diverge) over time;

THE SOLUTION:
− We need to detect these diverging clones, at 1000MLOC+ scale

SSCD:
A: Pre-processing source code
B: Fine-tuning the large language models (LLM)∗

C: Generating semantic vectors
D: Build index for corpus
E: Clone Retrieval

SSCD, with/without Remove-White-Space/Variable Anonymization 
Pre-processings, with Active Learning and with state-of-the-art LLMs

Max Recall at a precision of >=0.2 (across a Huawei code-base and 8 OS systems; 362 MLOC):

Preliminary results of trialling newer LLMs on the Huawei-provided Benchmark

Newer LLMs trialled on Huawei’s 480KLOC dataset

<table>
<thead>
<tr>
<th>LLM</th>
<th>Size (Parameters)</th>
<th>C F-score</th>
<th>C++ F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeT5</td>
<td>220M</td>
<td>85.04</td>
<td>90.45</td>
</tr>
<tr>
<td>CodeBERT</td>
<td>125M</td>
<td>77.16</td>
<td>83.33</td>
</tr>
<tr>
<td>GraphCodeBERT</td>
<td>125M</td>
<td>80.29</td>
<td>88.31</td>
</tr>
<tr>
<td>CuBERT</td>
<td>125M</td>
<td>87.14</td>
<td>88.6</td>
</tr>
<tr>
<td>CodeT5a</td>
<td>125M</td>
<td>90.20</td>
<td>97.04</td>
</tr>
<tr>
<td>SPT-Code</td>
<td>262M</td>
<td>97.84</td>
<td>92.77</td>
</tr>
</tbody>
</table>

Overall (and in Type 3 particularly), SSCD shows substantial improvement over CCFinderX:
− And pre-processings show further significant improvement:
− Preliminary indications suggest that incorporation of newer LLMs will improve things further.