两篇论文

SCIENCE AND TECHNOLOGY

《An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects》	2022	ESE
《Inferring Bug Signatures to Detect Real Bugs》	2022	TSE

GRADUATION DEFENSE

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

Whether the existing IRBL techniques are generalizable to industrial projects still needs to be studied.

Industrial Issues

- Software Product Line
- Mixture of Multiple Natural Languages
- Quality of bug reports

Noisy Bug Report 1
Summary: Code optimization for UDAP.
Description: Optimize the code for BRA and status inquiry in VAM module by reducing cyclomatic complexity.

Noisy Bug Report 2
Summary: fix lint warnings
Description: Several warnings still exist when inspecting the P300 library, needing to be fixed.

Fig. 3 Two noisy bug reports in Huawei projects

and 3)

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ1: ***How effective are IRBL techniques for industrial projects?***

RQ2: *How do industrial issues affect the effectiveness of IRBL techniques on industrial projects?*

RQ3: ***Can these issues be overcome?***

Table 1 The studied IRBL techniques

Technique	Venue	Year
BugLocator (Zhou et al. 2012)	Intl. Conf. on Software Engineering	2012
BLUiR (Saha et al. 2013)	Intl. Conf. on Automated Software Engineering	2013
AmaLgam (Wang and Lo 2014)	Intl. Conf. on Program Comprehension	2014
BRTracer (Wong et al. 2014)	Intl. Conf. on Software Maintenance and Evolution	2014
LearningToRank (Ye et al. 2014)	Symp. on the Foundations of Software Engineering	2014
Locus (Wen et al. 2016)	Intl. Conf. on Automated Software Engineering	2016

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ1: *How effective are IRBL techniques for industrial projects?*

Table 5 The performance on open-source Java projects

Projects	IRBL tech.	Top@1	Top@5	Top@10	MAP	MRR
ZXing	BugLocator	0.2000	0.6500	0.7500	0.3306	0.3837
	BLUiR	0.5000	0.6500	**0.8000**	0.4996	0.5903
	AmaLgam	**0.5500**	0.6500	0.7500	**0.5234**	**0.6143**
	BRTracer	0.3000	0.5500	0.7500	0.3937	0.4219
	LearnToRank	0.2277	0.5025	0.6683	0.2331	0.3646
	Locus	0.4500	**0.7000**	**0.8000**	0.4609	0.5551
SWT	BugLocator	0.3571	0.6939	0.7959	0.4458	0.5016
	BLUiR	**0.5408**	0.7551	0.8367	**0.5684**	**0.6480**
	AmaLgam	0.5306	0.7449	0.8367	0.5613	0.6341
	BRTracer	0.4694	**0.7857**	**0.8776**	0.5257	0.5967
	LearnToRank	0.4592	0.7449	0.8367	0.5214	0.5988
	Locus	0.5000	**0.7857**	0.8571	0.5463	0.6213
PDE	BugLocator	0.3729	0.6440	0.6780	0.3955	0.4961
	BLUiR	0.3390	0.5085	0.5932	0.3607	0.4377
	AmaLgam	0.3390	0.5593	0.6101	0.3669	0.4458
	BRTracer	**0.4407**	**0.7119**	**0.8146**	0.4248	**0.5473**
	LearnToRank	0.2078	0.5260	0.5974	0.2551	0.3533
	Locus	0.4000	0.6667	0.7500	**0.4321**	0.5274
AspectJ	BugLocator	0.2762	0.5140	0.6538	0.2299	0.3989
	BLUiR	0.2657	0.4895	0.5979	0.2215	0.3782
	AmaLgam	0.2622	0.5140	0.6224	0.2194	0.3840
	BRTracer	**0.3392**	**0.5874**	**0.7098**	**0.2477**	**0.4455**
	LearnToRank	0.2422	0.5026	0.6536	0.2327	0.3726
	Locus	0.2553	0.4680	0.5319	0.2412	0.3541
JDT	BugLocator	0.2021	0.3830	0.4681	0.2011	0.2932
	BLUiR	0.2660	0.4787	0.5745	0.2643	0.3775
	AmaLgam	**0.2766**	0.5000	0.5745	0.2567	0.3843
	BRTracer	**0.2766**	0.4681	0.5745	0.2900	0.3832
	LearnToRank	0.2021	0.3723	0.4894	0.2008	0.2847
	Locus	**0.2766**	**0.5426**	**0.5851**	**0.3202**	**0.3876**
Aggregate	BugLocator	0.2856	0.5424	0.6535	0.2844	0.4090
	BLUiR	0.3303	0.5421	0.6426	0.3146	0.4395
	AmaLgam	0.3303	0.5619	0.6552	0.3125	0.4429
	BRTracer	**0.3610**	**0.6142**	**0.7292**	0.3279	**0.4717**
	LearnToRank	0.2693	0.5257	0.6526	0.2805	0.3952
	Locus	0.3244	0.5660	0.6310	**0.3365**	0.4325

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ1: *How effective are IRBL techniques for industrial projects?*

Table 6 The performance on industrial projects (before eliminating noisy bug reports and without utilization of product information)

Projects	IRBL tech.	Top@1	Top@5	Top@10	MAP	MRR
NTA	BugLocator	0.3158	0.6667	0.8070	0.4403	0.4623
	BLUiR	0.3158	0.6140	0.7895	0.4368	0.4571
	AmaLgam	0.3158	0.6140	0.8070	0.4372	0.4574
	BRTracer	0.2982	0.6667	0.7895	0.4356	0.4525
	LearnToRank	**0.3509**	**0.7018**	**0.8421**	**0.4628**	**0.5137**
	Locus	**0.3509**	0.6316	0.8246	0.4606	0.4838
ESP	BugLocator	0.3781	0.6670	0.7668	0.4485	0.5049
	BLUiR	0.4031	0.6612	0.7735	0.4628	0.5203
	AmaLgam	0.4021	0.6747	0.7908	0.4704	0.5251
	BRTracer	0.3724	0.6651	0.7543	0.4429	0.4990
	LearnToRank	0.3426	**0.7140**	**0.8013**	0.4589	0.5086
	Locus	**0.5518**	0.7083	0.7793	**0.5927**	**0.6294**
DSLAM	BugLocator	**0.2197**	0.4005	0.4926	0.2224	0.3088
	BLUiR	0.2088	0.3934	0.4625	0.2109	0.3002
	AmaLgam	0.2153	0.4099	0.4816	0.2196	0.3116
	BRTracer	0.2175	**0.4230**	**0.5156**	**0.2263**	**0.3181**
	LearnToRank	0.1589	0.3551	0.4373	0.1843	0.2551
	Locus	0.1452	0.3145	0.3995	0.1702	0.2318
OLT	BugLocator	**0.2393**	**0.4456**	**0.5482**	**0.2454**	**0.3413**
	BLUiR	0.2272	0.4267	0.5292	0.2375	0.3284
	AmaLgam	0.2303	0.4382	0.5402	0.2412	0.3345
	BRTracer	0.2226	0.4416	0.5467	0.2351	0.3307
	LearnToRank	0.1946	0.4020	0.5074	0.2172	0.2979
	Locus	0.1822	0.3465	0.4421	0.2007	0.2673
IAS	BugLocator	0.1889	0.3496	0.4243	0.2309	0.2712
	BLUiR	0.1870	0.3731	0.4563	0.2342	0.2776
	AmaLgam	**0.2007**	**0.4008**	**0.4765**	**0.2515**	**0.2955**
	BRTracer	0.1875	0.3590	0.4431	0.2339	0.2742
	LearnToRank	0.1701	0.3346	0.4008	0.2137	0.2495
	Locus	0.1734	0.3228	0.4051	0.2143	0.2492
BSP	BugLocator	0.2253	0.4642	**0.5648**	0.2220	0.3367
	BLUiR	0.2218	0.4437	0.5512	0.2202	0.3289
	AmaLgam	0.2287	**0.4727**	0.5563	0.2377	0.3396
	BRTracer	0.2116	0.4505	0.5580	0.2180	0.3259
	LearnToRank	0.1621	0.4437	0.5410	0.2068	0.2856
	Locus	**0.2406**	0.4454	0.5307	**0.2641**	**0.3402**
ONT	BugLocator	**0.1960**	**0.4045**	0.4884	**0.2421**	**0.2954**
	BLUiR	0.1926	0.3979	0.4893	0.2379	0.2921
	AmaLgam	0.1935	0.4042	**0.4952**	0.2386	0.2930
	BRTracer	0.1945	0.4020	0.4930	0.2384	0.2932
	LearnToRank	0.1712	0.3703	0.4716	0.2247	0.2707
	Locus	0.1811	0.3647	0.4557	0.2286	0.2722

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ1: *How effective are IRBL techniques for industrial projects?*

Projects	IRBL tech.	Top@1	Top@5	Top@10	MAP	MRR
ANP	BugLocator	0.1774	0.4229	0.5111	0.2147	0.2900
	BLUiR	**0.1806**	0.4222	0.5056	0.2141	0.2906
	AmaLgam	0.1788	0.4240	0.5074	0.2151	0.2896
	BRTracer	0.1785	0.4309	0.5281	0.2171	**0.2934**
	LearnToRank	0.1567	**0.4465**	**0.5303**	**0.2177**	0.2893
	Locus	0.1748	0.3544	0.4603	0.2056	0.2658
UTS	BugLocator	**0.1734**	**0.3822**	0.4687	**0.2006**	**0.2735**
	BLUiR	0.1721	0.3756	0.4590	0.1964	0.2700
	AmaLgam	0.1721	0.3738	0.4590	0.1974	0.2709
	BRTracer	**0.1734**	0.3791	**0.4704**	0.2000	0.2731
	LearnToRank	0.1249	0.3517	0.4373	0.1706	0.2276
	Locus	0.1377	0.2992	0.3839	0.1640	0.2192
WDM	BugLocator	0.1873	**0.4508**	**0.5603**	**0.2126**	**0.3105**
	BLUiR	0.1873	0.4296	0.5324	0.2027	0.3017
	AmaLgam	**0.1888**	0.4330	0.5360	0.2059	0.3048
	BRTracer	0.1835	0.4395	0.5461	0.2043	0.3042
	LearnToRank	0.1855	0.4200	0.5378	0.2119	0.2982
	Locus	0.1724	0.4060	0.5191	0.2025	0.2837
Aggregate	BugLocator	**0.2110**	**0.4314**	0.5263	**0.2370**	**0.3162**
	BLUiR	0.2077	0.4219	0.5156	0.2321	0.3110
	AmaLgam	0.2105	0.4311	0.5240	0.2369	0.3161
	BRTracer	0.2051	0.4307	**0.5287**	0.2328	0.3129
	LearnToRank	0.1802	0.4071	0.5039	0.2208	0.2887
	Locus	0.1893	0.3703	0.4656	0.2196	0.2798

Finding 1. The state-of-the-art IRBL techniques tend to have an excellent performance when applied to small-scale projects. However, the performance degrades on large-scale projects. Considering the large-scale projects only, IRBL techniques yield lower performance on industrial projects.
Finding 2. The performance is stable across industrial large-scale projects, and the differences in performance among IRBL techniques is non-obvious.

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ2: *How do industrial issues affect the effectiveness of IRBL techniques on industrial projects?*

Lexical similarity feature

$$FinalScore = (1 - \alpha) \times N(Score_L) + \alpha \times N(Score_C)$$

The lexical similarity between bug reports and source code files has been regarded as the crucial feature in IRBL techniques.

Collaborative filtering feature

For a specific bug report, the collaborative filtering score of a source code file is calculated based on the textual similarities of their relevant historical bug reports and the number of files that are modified to fix each bug report.

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ2: *How do industrial issues affect the effectiveness of IRBL techniques on industrial projects?*

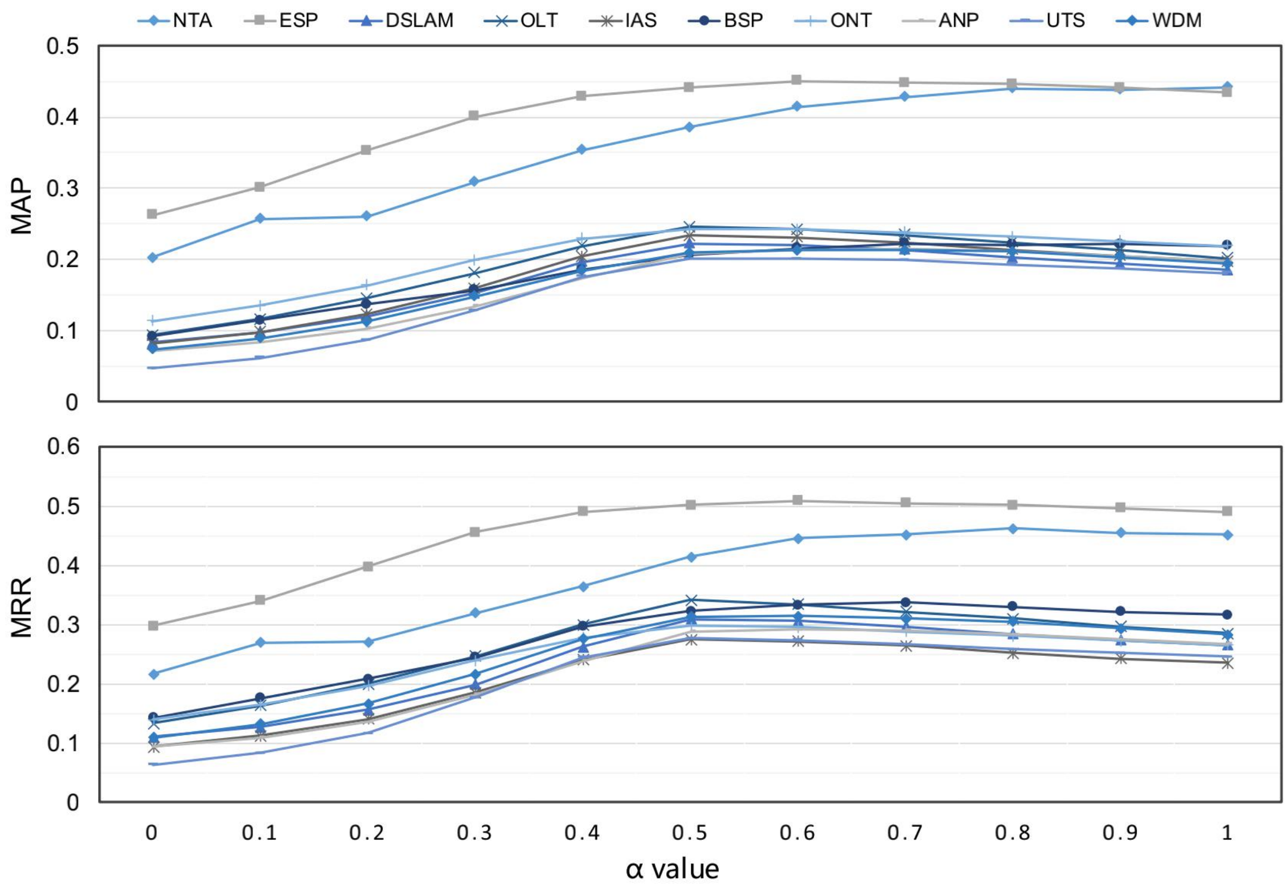


Fig. 4 Impact of varying α in BugLocator (α represents the weight of collaborative filtering feature)

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ2: *How do industrial issues affect the effectiveness of IRBL techniques on industrial projects?*

Table 7 The performance of BugLocator for non-SPL projects and SPL projects

Projects	Top@1	Top@5	Top@10	MAP	MRR
Non-SPL projects					
DSLAM	0.2197	0.4005	0.4926	0.2224	0.3088
OLT	0.2393	0.4456	0.5482	0.2454	0.3413
BSP	0.2253	0.4642	0.5648	0.2220	0.3367
UTS	0.1734	0.3822	0.4687	0.2006	0.2735
Aggregate	0.2214	0.4256	0.5230	0.2308	0.3213
SPL projects					
IAS	0.1889	0.3496	0.4243	0.2309	0.2712
ONT	0.1960	0.4045	0.4884	0.2421	0.2954
ANP	0.1774	0.4229	0.5111	0.2147	0.2900
WDM	0.1873	0.4508	0.5603	0.2126	0.3105
Aggregate	0.1876	0.4156	0.5080	0.2237	0.2955

An empirical study of the effectiveness of IR-based bug localization for large-scale industrial projects

2022 ESE

RQ3: *Can these issues be overcome?*

Table 8 Comparison between performance results for BugLocator including and excluding noisy bug reports

Projects	Noisy bugs included		Noisy bugs excluded	
	MAP	MRR	MAP	MRR
NTA	0.4403	0.4623	↘ 0.4328	↘ 0.4617
ESP	0.4485	0.5049	↗ 0.4930**	↗ 0.5462**
DSLAM	0.2224	0.3088	↗ 0.2594**	↗ 0.3283*
OLT	0.2454	0.3413	↗ 0.2699**	↗ 0.3534**
IAS	0.2309	0.2712	↗ 0.2496	↗ 0.2830
BSP	0.2220	0.3367	↘ 0.2212	↘ 0.3317
ONT	0.2421	0.2954	↗ 0.2591*	↗ 0.3111*
ANP	0.2147	0.2900	↗ 0.2291*	↗ 0.3040*
UTS	0.2006	0.2735	↗ 0.2037	↗ 0.2746
WDM	0.2126	0.3105	↗ 0.2144	↗ 0.3178
Aggregate	0.2370	0.3162	↗ 0.2515**	↗ 0.3272**

$*$: p-value < 0.05, $**$: p-value < 0.01, ↗: increased, ↘: decreased

Inferring Bug Signatures to Detect Real Bugs

2022 TSE

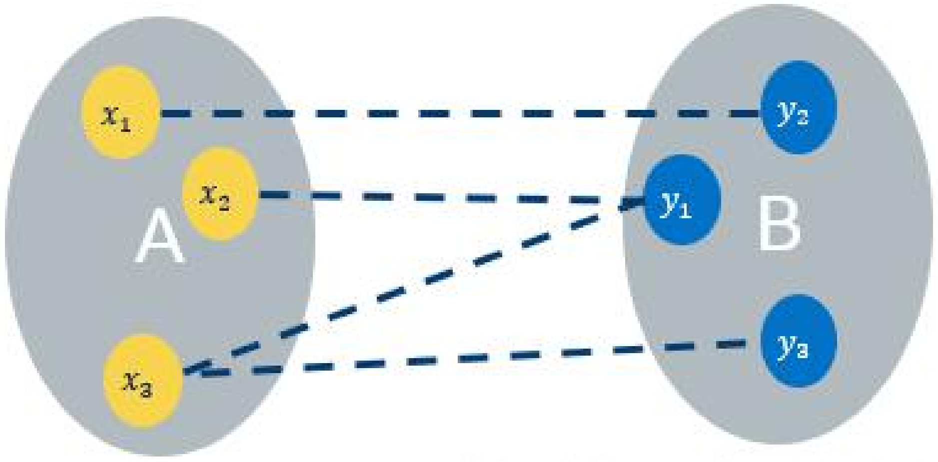
Bug Signature

A bug signature is ***a set of program elements that explain the cause or the effect of a bug***, and a bug signature can be easily translated to bug patterns of existing static tools.

Hungarian algorithm

Hungarian algorithm is a classical algorithm that solves the assignment problem.

匈牙利算法是一种在多项式时间内求解任务分配问题的组合优化算法。



Inferring Bug Signatures to Detect Real Bugs

2022 TSE

Motivation Example

```java
protected ... getWriteDirectory(long writeSize){
  ...
  directory = getDirectories().getWriteableLocation(...);
  if (directory == null)
    throw new RuntimeException("Insufficient disk space to
          write " + writeSize + " bytes");
  return directory;
}
```

(a) The buggy code

```java
protected ... getWriteDirectory(long writeSize){
  ...
  directory = getDirectories().getWriteableLocatic
  if (directory == null)
    throw new FSWriteError(new IOException("Insuf
          disk space to write " + writeSize + " by
          ");
  return directory;
}
```

(b) The fixed code

Fig. 1. CASSANDRA-11448.

invokevirtual Lorg/apache/Cassandra/io/util/DiskAwareRunnable, getDirectories()

↓

invokevirtual Lorg/apache/cassandra/db/Directories, getWriteableLocation(...)

↓

invokespecial Ljava/lang/RuntimeException, <init>(...)

(a) The inferred bug signature

```java
public ... getWriteDirectory(...) {
  ...
  d = getDirectories().getWriteableLocation(...);
  if (d == null)
    throw new RuntimeException(...);
  return d;
}
```

(b) The buggy code in DirObjectFactoryHelper.java

Fig. 2. Our reported bug.

Inferring Bug Signatures to Detect Real Bugs

2022 TSE

node：method invocation
edge： a control or data dependency between two nodes

GRAPA

- Construction of Mappings
- Inferring Bug Signatures

- ➢ *Levenshtein edit distance*
- ➢ *Hungarian algorithm*

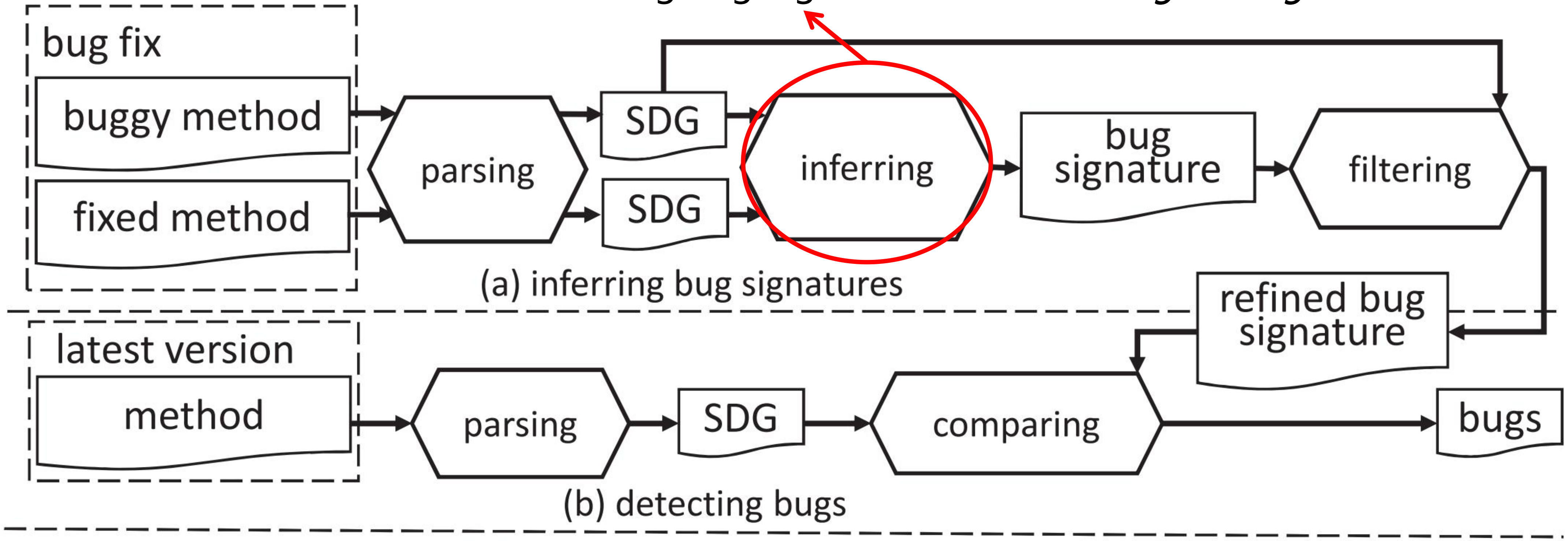


Fig. 3. The overview of DEPA.

南京航空航天大学
Nanjing University of Aeronautics and Astronautics

2022

THANK YOU

THNAK YOU

GRADUATION DEFENSE