Borda count :

根据IR方法生成的排名对检索到的link进行赋值。假设有方法集合C，每个方法Ci对link k为的排名设为ri,k；Mi表示被Ci检索到的link并且分数不为0的总数。
那么，对于link k来说他的borda计数就是 Mi - ri,k；所以total为之和。

Score Addition:

将每个IR方法结果相加

Ri,k	C1	C2	C3	Mi-ri,k
Link1	1	2	5	(5-1)+(5-2)+(5-3)=3
Link2	2	4	3	...=6
Link3	3	2	1	9
总共有5个 link				

《A qualitative reasoning approach to spectrum-based fault localization》	2018	ICSE
《Leveraging Qualitative Reasoning to Improve SFL》	2018	IJCAI
《A Test-suite Diagnosability Metric for Spectrum-based Fault Localization Approaches》	2017	ICSE
《 A Theoretical and Empirical Analysis of Program Spectra Diagnosability》	2019	TSE
《Demystifying the Combination of Dynamic Slicing and Spectrum-based Fault Localization》	2019	IJCAI
《The Mutation and Injection Framework-Evaluating Clone Detection Tools with Mutation Analysis》	2019	TSE

Rui Abreu ; Alexandre Perez

Demystifying the Combination of Dynamic Slicing and Spectrum-based Fault Localization

作者：Sofia Reis , Rui Abreu and Marcelo d’ Amorim

汇报人：陈冰婷　　　导师：邹卫琴

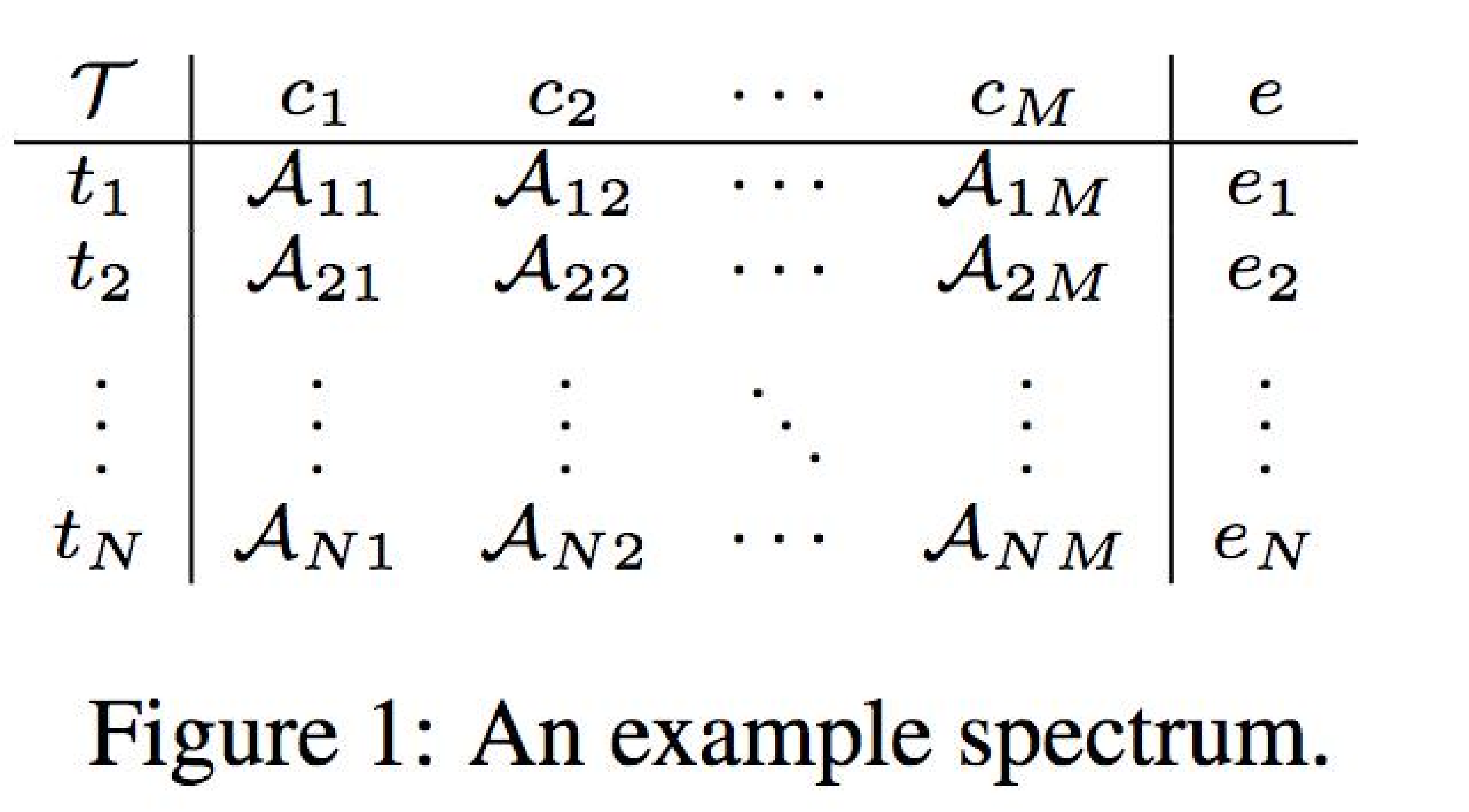
目录
Contents

PART 01 Introduction

PART 02 Experiments

PART 03 Performance

SFL

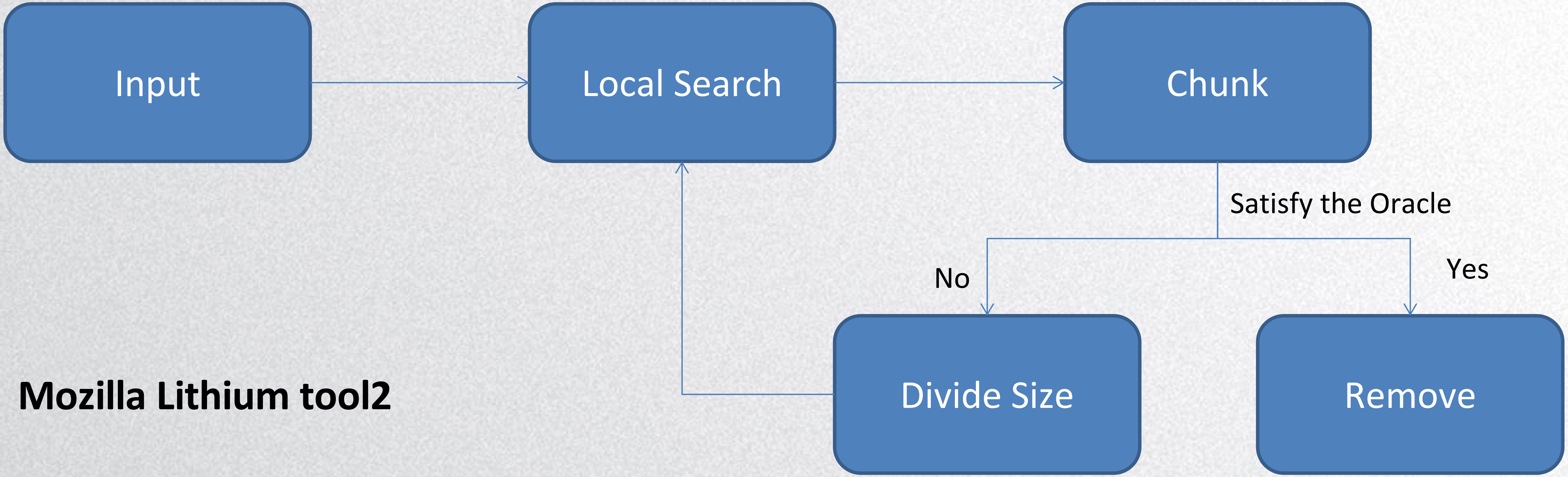


$\mathcal{T}$	c_1	c_2	$\cdots$	c_M	e
t_1	$\mathcal{A}_{11}$	$\mathcal{A}_{12}$	$\cdots$	$\mathcal{A}_{1M}$	e_1
t_2	$\mathcal{A}_{21}$	$\mathcal{A}_{22}$	$\cdots$	$\mathcal{A}_{2M}$	e_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
t_N	$\mathcal{A}_{N1}$	$\mathcal{A}_{N2}$	$\cdots$	$\mathcal{A}_{NM}$	e_N

Figure 1: An example spectrum.

DS

Oracle Definition： ***the test produces the same failure manifestation as the one observed with the test execution on the original program.***



Mozilla Lithium tool2

Tendem-FL

01 **Compute spectra S and ranking R for the input test suite;**

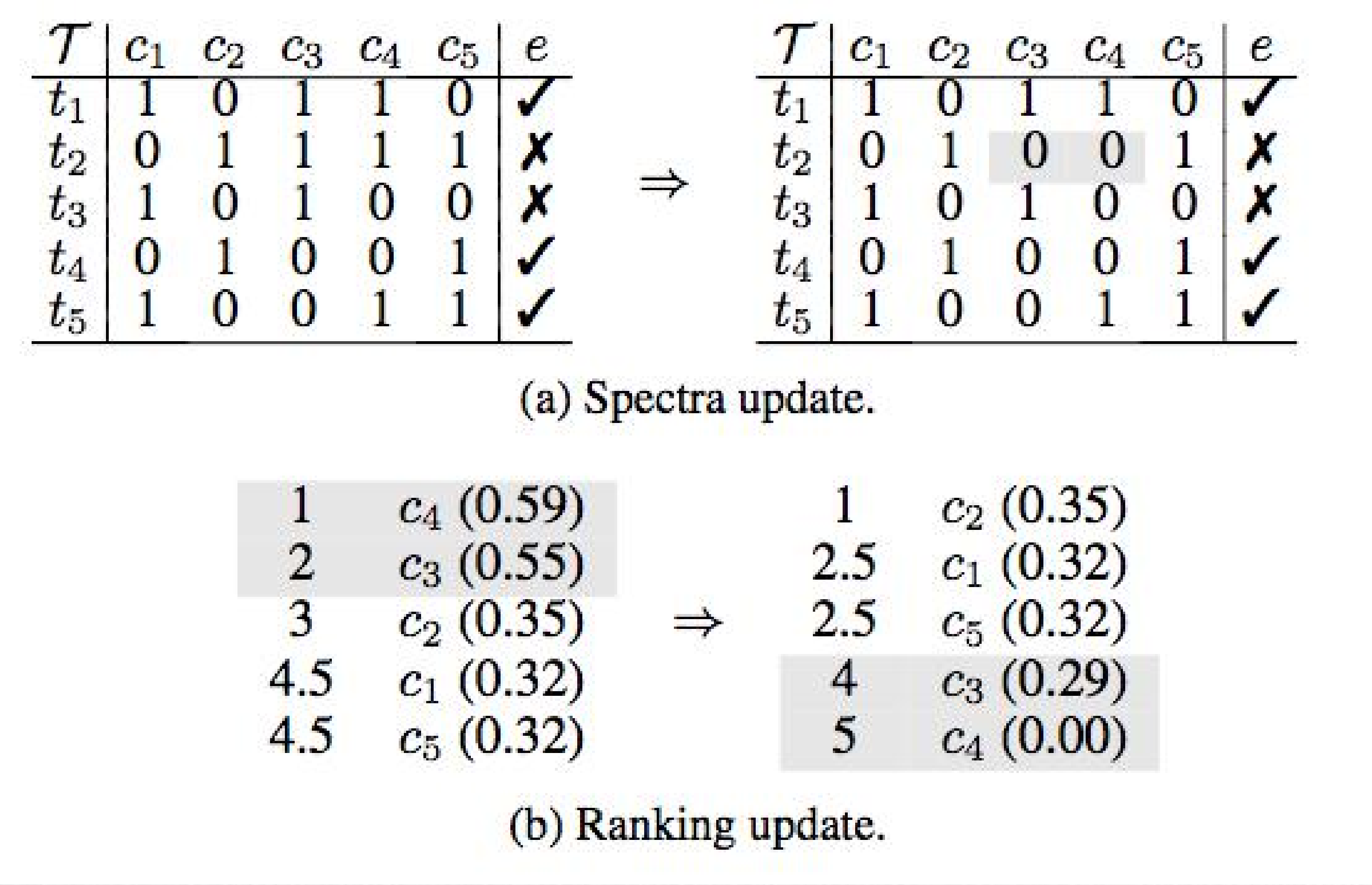
02 **Select top k most suspicious classes, according to R;**

03 **Compute slicer for every suspicious file, obtained in Step 2, and every failing test;**

04 **Assemble all the resulting slicers of each failing test;**

05 **Adjust spectra S, from step 1, with the slices of each failing test, from step 4, and then recompute the ranking.**

Tendem-FL



$\mathcal{T}$	c_1	c_2	c_3	c_4	c_5	e
t_1	1	0	1	1	0	✓
t_2	0	1	1	1	1	✗
t_3	1	0	1	0	0	✗
t_4	0	1	0	0	1	✓
t_5	1	0	0	1	1	✓

⇒

$\mathcal{T}$	c_1	c_2	c_3	c_4	c_5	e
t_1	1	0	1	1	0	✓
t_2	0	1	0	0	1	✗
t_3	1	0	1	0	0	✗
t_4	0	1	0	0	1	✓
t_5	1	0	0	1	1	✓

(a) Spectra update.

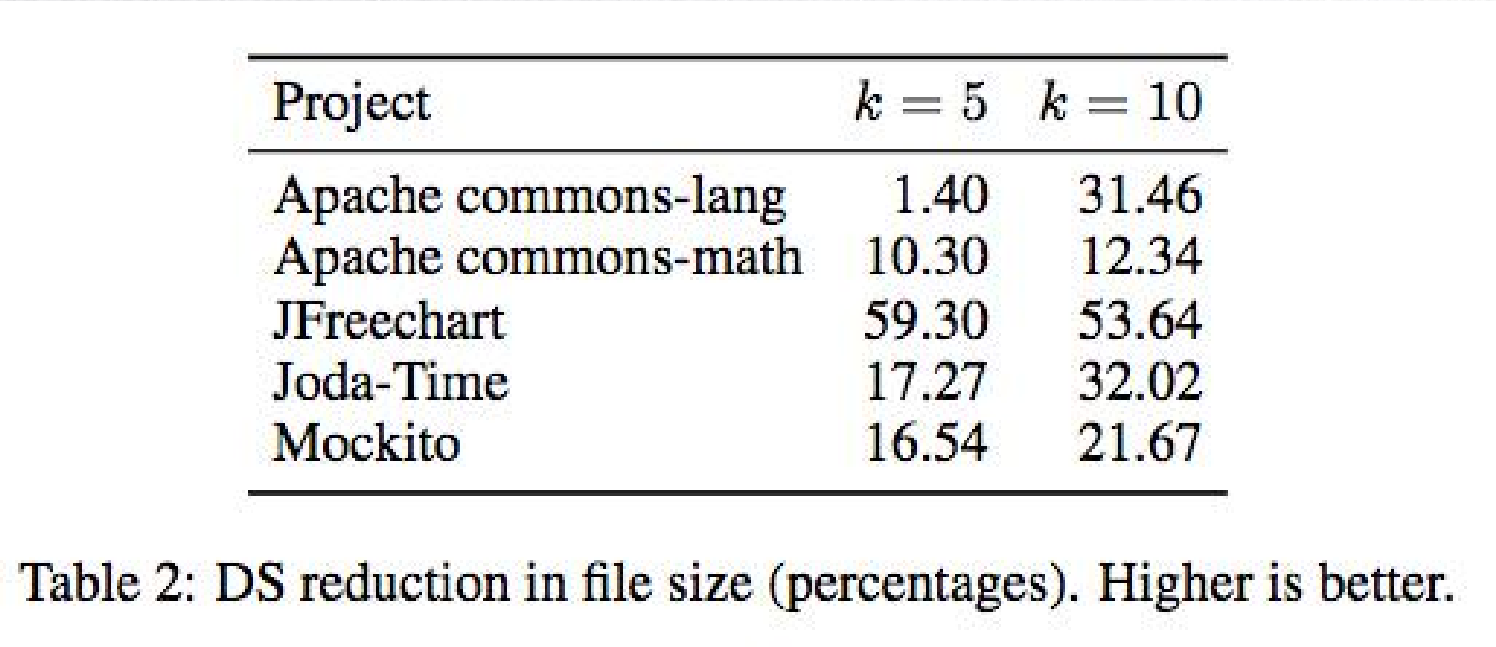
1	c_4 (0.59)
2	c_3 (0.55)
3	c_2 (0.35)
4.5	c_1 (0.32)
4.5	c_5 (0.32)

⇒

1	c_2 (0.35)
2.5	c_1 (0.32)
2.5	c_5 (0.32)
4	c_3 (0.29)
5	c_4 (0.00)

(b) Ranking update.

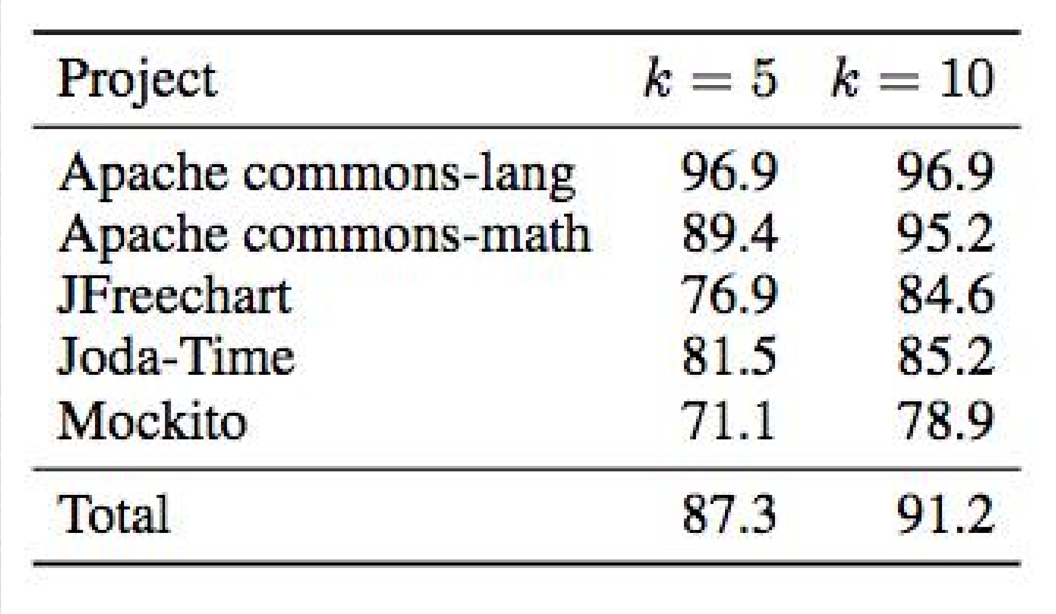
RQ1: How effective is DS in eliminating code?



Project	$k = 5$	$k = 10$
Apache commons-lang	1.40	31.46
Apache commons-math	10.30	12.34
JFreechart	59.30	53.64
Joda-Time	17.27	32.02
Mockito	16.54	21.67

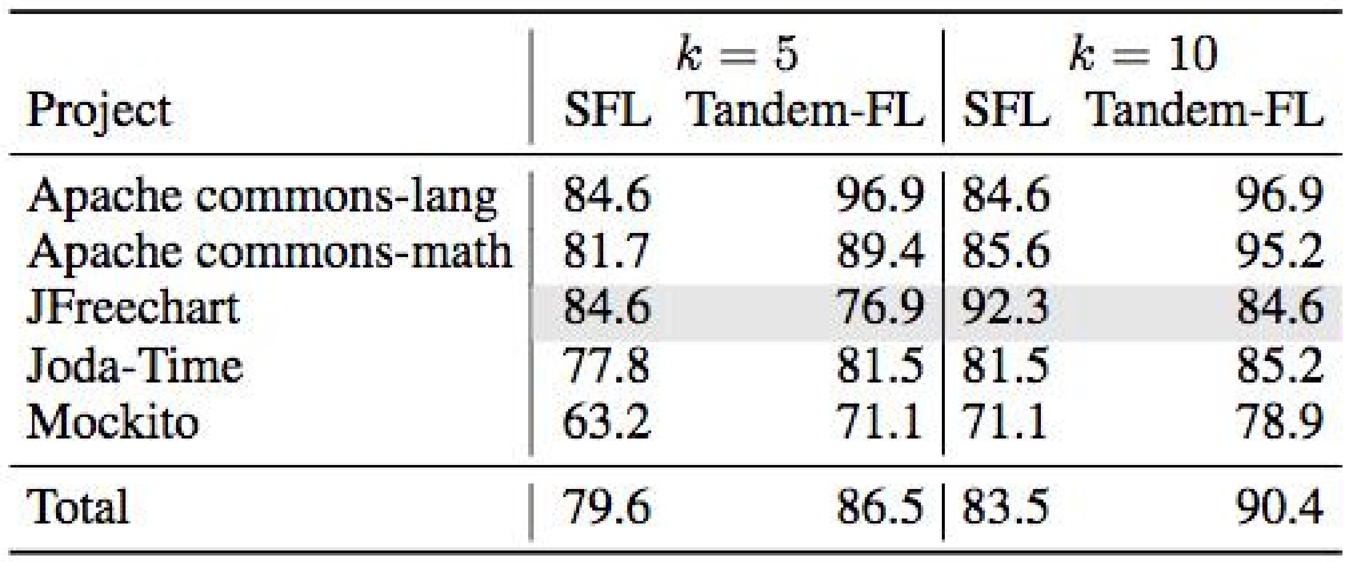
Table 2: DS reduction in file size (percentages). Higher is better.

RQ2: How often does DS miss faulty statements?



Project	$k = 5$	$k = 10$
Apache commons-lang	96.9	96.9
Apache commons-math	89.4	95.2
JFreechart	76.9	84.6
Joda-Time	81.5	85.2
Mockito	71.1	78.9
Total	87.3	91.2

Table 3: Tandem-FL^k performance on capturing faulty statements, as percentages. Higher is better.



	$k = 5$		$k = 10$	
Project	SFL	Tandem-FL	SFL	Tandem-FL
Apache commons-lang	84.6	96.9	84.6	96.9
Apache commons-math	81.7	89.4	85.6	95.2
JFreechart	84.6	76.9	92.3	84.6
Joda-Time	77.8	81.5	81.5	85.2
Mockito	63.2	71.1	71.1	78.9
Total	79.6	86.5	83.5	90.4

Table 4: Number of faults where at least one of the faulty statements appears at the report of the technique. Higher is better.

RQ3: How effective is Tandem-FL for bug localization?

$$\Delta C = C(\text{SFL}) - C(\text{Tandem-FL})$$

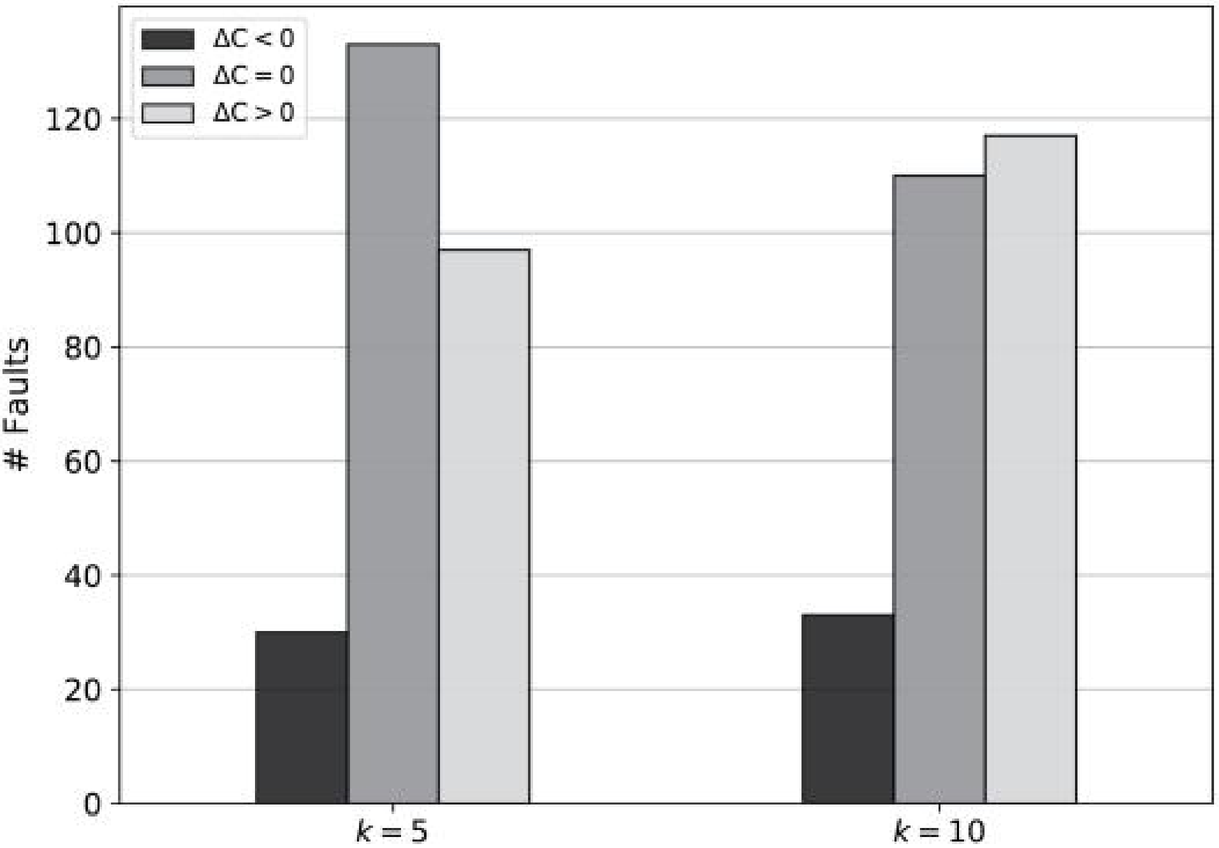


Figure 3: Delta Cost of Diagnosis (ΔC) per k

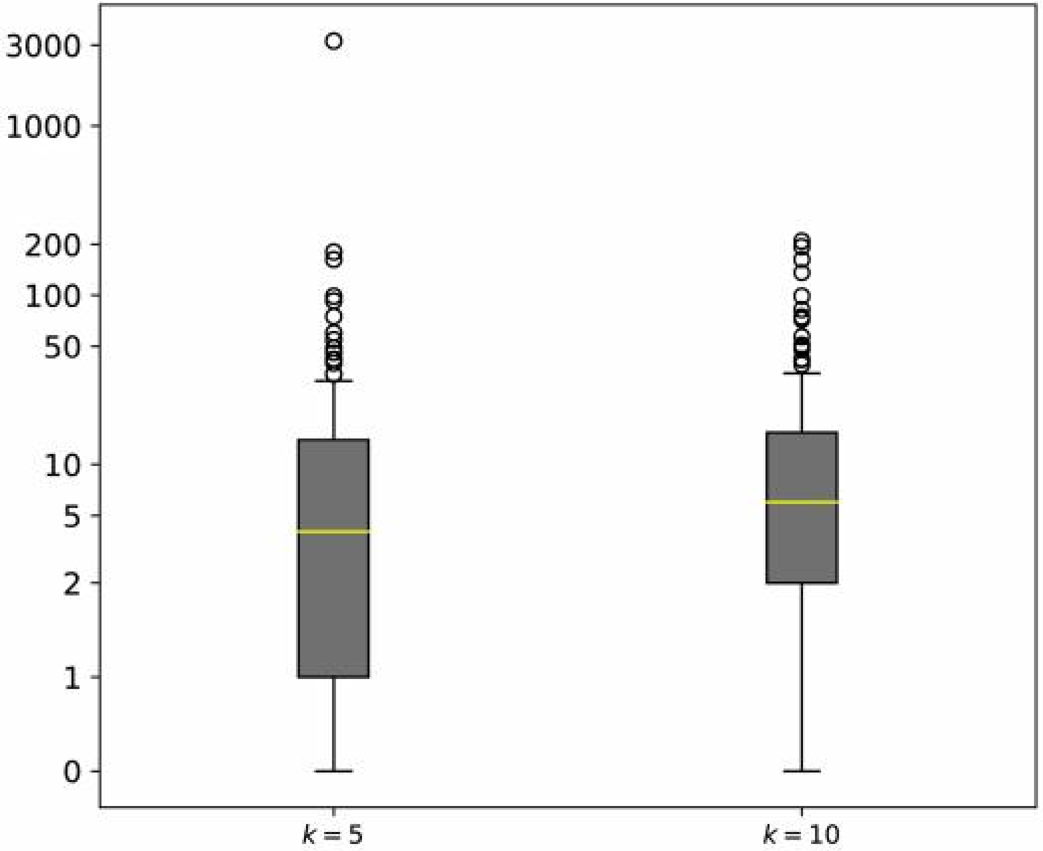
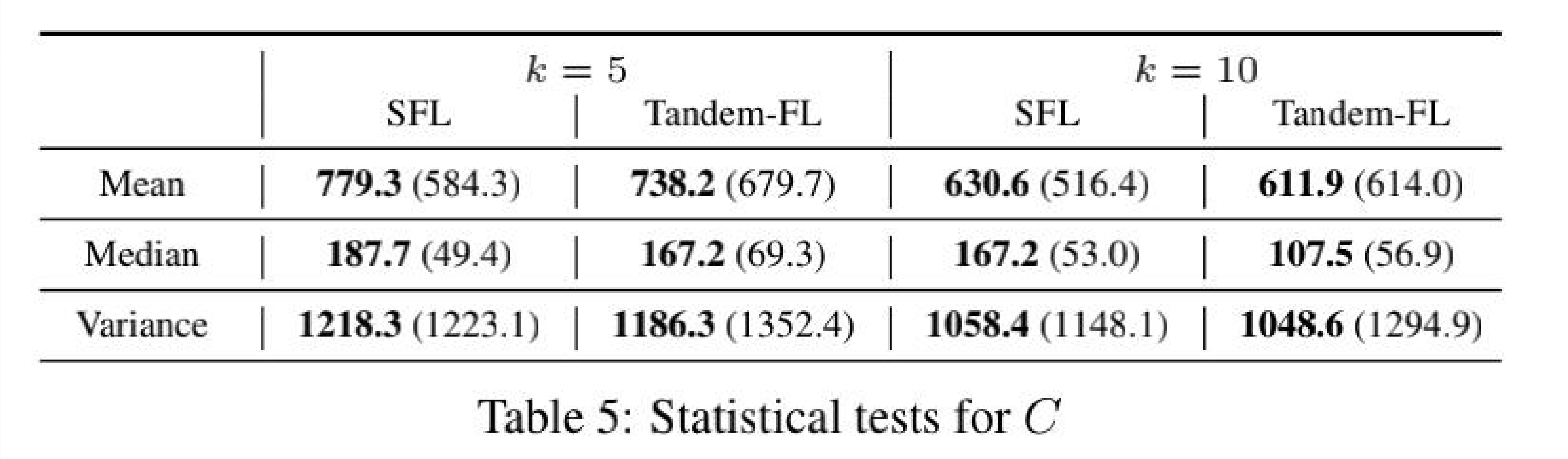


Figure 4: Distributions of ΔC considering all cases where Tandem-FL^k outperformed the baseline

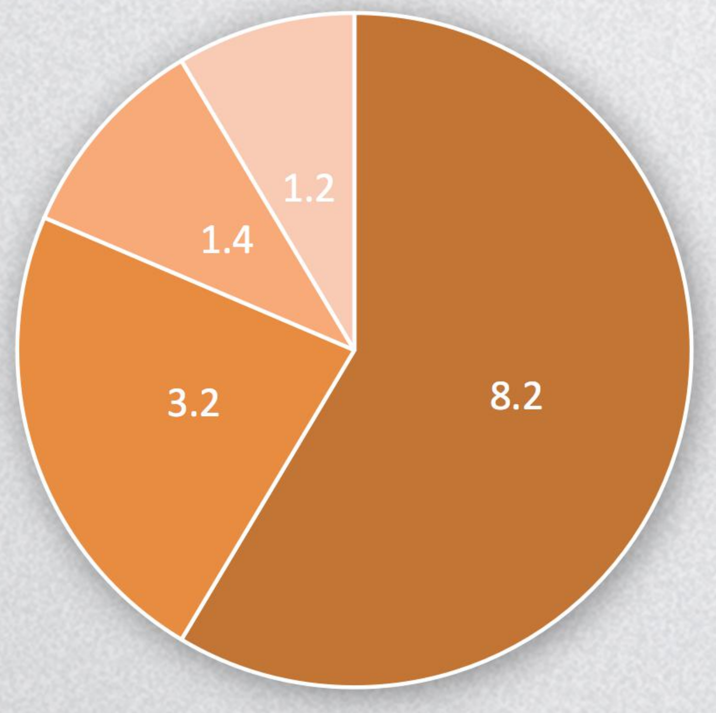
RQ3: How effective is Tandem-FL for bug localization?



	$k = 5$		$k = 10$	
	SFL	Tandem-FL	SFL	Tandem-FL
Mean	**779.3** (584.3)	**738.2** (679.7)	**630.6** (516.4)	**611.9** (614.0)
Median	**187.7** (49.4)	**167.2** (69.3)	**167.2** (53.0)	**107.5** (56.9)
Variance	**1218.3** (1223.1)	**1186.3** (1352.4)	**1058.4** (1148.1)	**1048.6** (1294.9)

Table 5: Statistical tests for C

Results

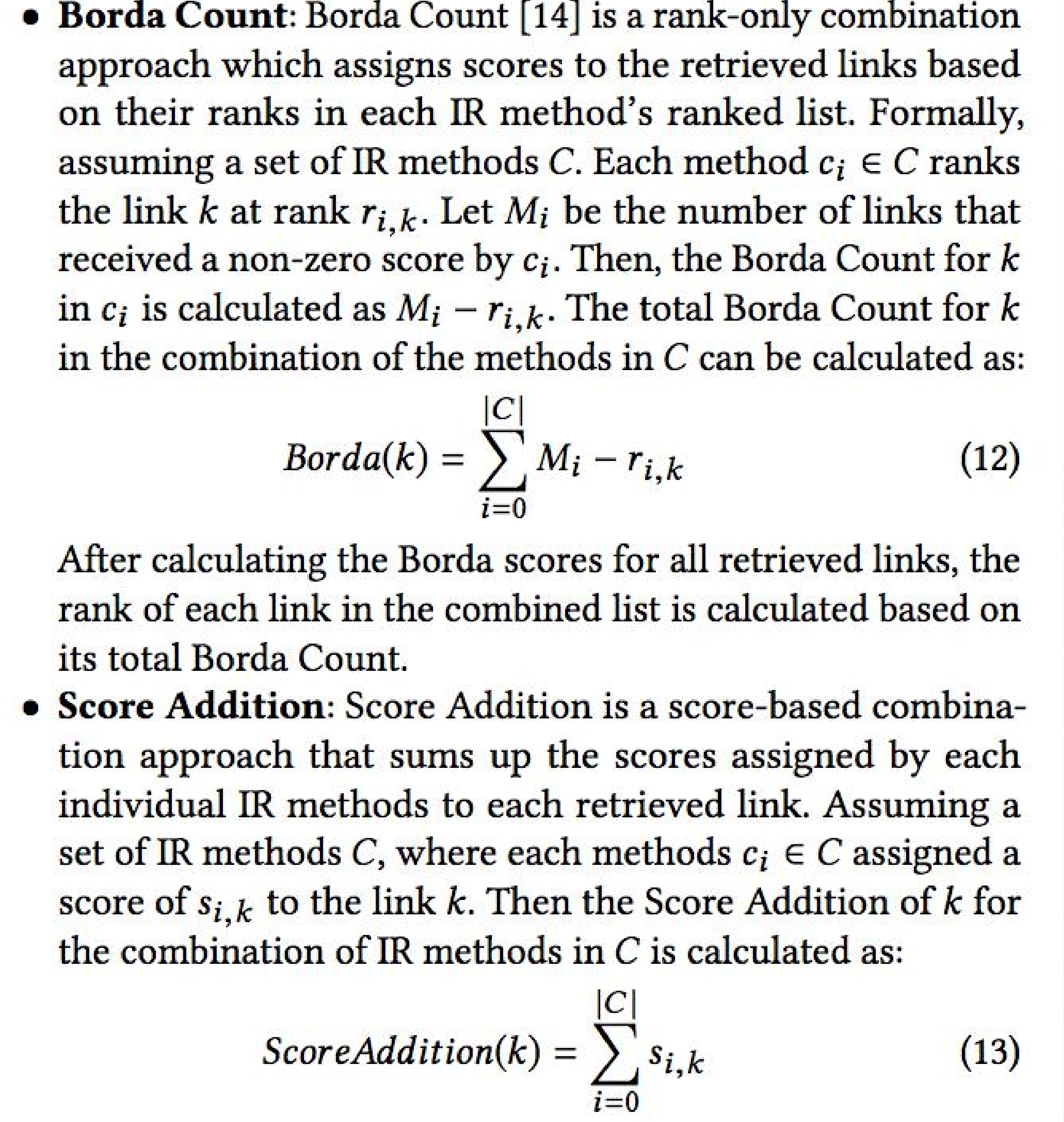


- DS misses faulty statements infrequently 9% (23 misses in 260 cases)

- The DS-SFL combination, coined as Tandem-FLk, improves the diagnostic accuracy up to 73.7% (13.4% on average).

THANK YOU FOR YOUR LISTENING.

谢谢您的聆听



- **Borda Count**: Borda Count [14] is a rank-only combination approach which assigns scores to the retrieved links based on their ranks in each IR method's ranked list. Formally, assuming a set of IR methods C. Each method $c_i \in C$ ranks the link k at rank $r_{i,k}$. Let M_i be the number of links that received a non-zero score by c_i. Then, the Borda Count for k in c_i is calculated as $M_i - r_{i,k}$. The total Borda Count for k in the combination of the methods in C can be calculated as:

$$Borda(k) = \sum_{i=0}^{|C|} M_i - r_{i,k} \qquad (12)$$

  After calculating the Borda scores for all retrieved links, the rank of each link in the combined list is calculated based on its total Borda Count.

- **Score Addition**: Score Addition is a score-based combination approach that sums up the scores assigned by each individual IR methods to each retrieved link. Assuming a set of IR methods C, where each methods $c_i \in C$ assigned a score of $s_{i,k}$ to the link k. Then the Score Addition of k for the combination of IR methods in C is calculated as:

$$ScoreAddition(k) = \sum_{i=0}^{|C|} s_{i,k} \qquad (13)$$