

Path Projection

For User-Centered Static Analysis Tools

Khoo Yit Phang, Jeff Foster, Michael Hicks, Vibha Sazawal

University of Maryland

PASTE 2008, November 10

Success in Static Analysis

- Coverity, Fortify, Grammatech, Klocwork, many others are selling static analysis tools
- Microsoft, Mozilla, and others are integrating static analysis into development
- Very active static analysis research community

But there's a problem...

- Research has focused on static analysis **algorithms**
- But, programmers use **tools**, not algorithms
- Static analysis tools are only useful if programmers can understand the results
- Our goal: develop ways to make static analysis tools more **user-centered**

Path Projection

- A new UI toolkit for visualizing **program paths**
 - call stacks and control-flow paths
- Paths are very common in static analysis tool output
 - Helping users understand paths will help many static analysis tools
 - We have applied Path Projection to Locksmith and BLAST
- Experimental evaluation
 - Task: triaging Locksmith error reports
 - Result: 18% improvement in completion time, similar accuracy

Case Study: Locksmith

Polyvios Pratikakis et al. (PLDI 2006)

- Static data race detector for C
 - Data race: Two or more threads access a shared variable at the same time
- Locksmith reports call stacks to possibly-racing dereferences
 - To triage, user must decide whether multiple paths are **simultaneously realizable**

Locksmith in Standard Viewer

The image shows a screenshot of the Standard Viewer application, a tool used for analyzing program execution paths. The main window displays the source code of a file named `main.c`. The code includes several standard C headers and defines a `main` function. A callout box highlights the text: "Standard Viewer designed to mimic typical editors/IDEs".

On the right side, there is a "Find" panel with a search input field and "Find" and "Cancel" buttons. Below it is a "Path Report" panel showing a warning about a possible data race. The report lists two execution paths:

- 1. `<in main.c>`
`main():42` -> `pthread_create()`
`<in Signal.c>`
`signal_waiter():36`
`sigarm_handler():61`
`<in Misc.c>`
`updateProgressBar():193` -> dereference
locks: -
- 2. `<in main.c>`
`main():117`
`<in Aget.c>`
`resume_get():171` -> `pthread_create()`
`<in Download.c>`
`http_get():121`
`<in Misc.c>`
`updateProgressBar():193` -> dereference
locks: -

Standard Viewer designed to
mimic typical editors/IDEs

Locksmith in Standard Viewer

The image shows a screenshot of the Path Visualizer application. The main window displays the source code of a file named `main.c`. The code includes several standard C headers and custom headers, followed by a `main` function. The `main` function contains several external declarations and a block of code with comments. The right-hand side of the application shows a 'Find' dialog box with a search field and 'Find' and 'Cancel' buttons. Below the search field is a 'Path Report' section. The report contains a warning about a possible data race of `prev` in `Misc.c:<global>:15`. It lists two paths leading to the error: 1. `<in main.c>` `main():42` `-> pthread_create()` `<in Signal.c>` `signal_waiter():36` `sigarm_handler():61` `<in Misc.c>` `updateProgressBar():193` `-> dereference`. 2. `<in main.c>` `main():117` `<in Aget.c>` `resume_get():171` `-> pthread_create()` `<in Download.c>` `http_get():121` `<in Misc.c>`. The report also shows 'locks: -'. A pink callout box at the bottom right contains the text 'Locksmith error report with hyperlinks'.

```
main.c
1  #include <unistd.h>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5  #include <signal.h>
6  #include <pthread.h>
7
8
9  #include "Defs.h"
10 #include "Data.h"
11 #include "Misc.h"
12 #include "Aget.h"
13 #include "Signal.h"
14 #include "Resume.h"
15 #include "main.h"
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28      * struct request stores all the information that might be
29      * of interest
```

Find ▶ Help
Find Cancel

Path Report

Warning: Possible data race of `prev` (`Misc.c:<global>:15`)
at:

- `<in main.c>`
`main():42` `-> pthread_create()`
`<in Signal.c>`
`signal_waiter():36`
`sigarm_handler():61`
`<in Misc.c>`
`updateProgressBar():193` `-> dereference`
- `<in main.c>`
`main():117`
`<in Aget.c>`
`resume_get():171` `-> pthread_create()`
`<in Download.c>`
`http_get():121`
`<in Misc.c>`

locks: -

Locksmith error report with hyperlinks

Find[▶ Help](#)

Find

Cancel

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>

main():42 -> pthread_create()

<in Signal.c>

signal_waiter():36

sigarm_handler():61

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

2. <in main.c>

main():117

<in Aget.c>

resume_get():171 -> pthread_create()

<in Download.c>

http_get():121

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

Locksmith Error Report

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Locksmith Error Report

Shared variable

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>

main():42 -> pthread_create()

<in Signal.c>

signal_waiter():36

sigarm_handler():61

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

2. <in main.c>

main():117

<in Aget.c>

resume_get():171 -> pthread_create()

<in Download.c>

http_get():121

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

Locksmith Error Report

Shared variable

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

```
1. <in main.c>  
main():42 -> pthread_create()  
<in Signal.c>  
signal_waiter():36  
sigarm_handler():61  
<in Misc.c>  
updateProgressBar():193 -> dereference
```

locks: -

Call stacks
leading to
race

```
2. <in main.c>  
main():117  
<in Aget.c>  
resume_get():171 -> pthread_create()  
<in Download.c>  
http_get():121  
<in Misc.c>  
updateProgressBar():193 -> dereference
```

locks: -

Locksmith Error Report

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>

main():42 -> pthread_create()

<in Signal.c>

signal_waiter():36

sigarm_handler():61

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

2. <in main.c>

main():117

<in Aget.c>

resume_get():171 -> pthread_create()

<in Download.c>

http_get():121

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

Thread creation

Locksmith Error Report

Path Report

Warning: Possible data race of
`prev` (Misc.c:<global>:15)

at:

1. <in main.c>

`main():42` -> `pthread_create()`

<in Signal.c>

`signal_waiter():36`

`sigarm_handler():61`

<in Misc.c>

`updateProgressBar():193` -> `dereference`

locks: -

2. <in main.c>

`main():117`

<in Aget.c>

`resume_get():171` -> `pthread_create()`

<in Download.c>

`http_get():121`

<in Misc.c>

`updateProgressBar():193` -> `dereference`

locks: -

Thread creation

Dereference

Locksmith Error Report

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>

main():42 -> pthread_create()

<in Signal.c>

signal_waiter():36

sigarm_handler():61

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

2. <in main.c>

main():117

<in Aget.c>

resume_get():171 -> pthread_create()

<in Download.c>

http_get():121

<in Misc.c>

updateProgressBar():193 -> dereference

locks: -

Thread creation

Dereference

w/no locks held

Locksmith Error Report

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Thread creation

Dereference

w/no locks held

Locksmith Error Report

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)

at:

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

The image shows a screenshot of the Path Visualizer application. The main window displays the source code for `main.c`. The code includes several system headers and local headers, followed by a `main` function. The `main` function contains several `pthread_create` calls. A `Find` window is open on the right, showing a `Path Report` for a warning about a possible data race. The report lists two call stacks:

```
Warning: Possible data race of
prev (Misc.c:<global>:15)
at:
1. <in main.c>
   main():42      -> pthread_create()
   <in Signal.c>
   signal_waiter():36
   sigarm_handler():61
   <in Misc.c>
   updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
   main():117
   <in Aget.c>
   resume_get():171 -> pthread_create()
   <in Download.c>
   http_get():121
   <in Misc.c>
```

The `main.c` code is as follows:

```
1  #include <unistd.h>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5  #include <signal.h>
6  #include <pthread.h>
7
8
9  #include "Defs.h"
10 #include "Data.h"
11 #include "Misc.h"
12 #include "Aget.h"
13 #include "Signal.h"
14 #include "Resume.h"
15 #include "main.h"
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28      * struct request stores all the information that might be
29      * of interest
```

Triage: are these call stacks simultaneously realizable?

Triaging Locksmith

Path Visualizer

main.c split close

```
main.c
1  #include <unistd.h>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5  #include <signal.h>
6  #include <pthread.h>
7
8
9  #include "Defs.h"
10 #include "Data.h"
11 #include "Misc.h"
12 #include "Aget.h"
13 #include "Signal.h"
14 #include "Resume.h"
15 #include "main.h"
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28      * struct request stores all the information that might be
29      * of interest
```

Find Help

Find Cancel

Path Report

Warning: Permission denied
prev (M
at:

Begin by clicking

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

Path Visualizer

main.c split close

```
main.c
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc, argv, "p:l:n:hfv")) != -1) {
48     switch(c) {
49         case 'p':
50             req->port = atoi(optarg);
51             break;
52         case 'f':
53             fsuggested = 1;
54             break;
55         case 'l':
56             strncpy(req->lfile, optarg, MAXBUFSIZ);
57             break;
58         case 'n':
59             if ((nthreads = atoi(optarg)) > MAXTHREADS) {
60                 Log("Error: Maximum # of threads allowed is %
61                 nthreads = 0;
62             }
63             break;
64         case 'h':
65             printf("%s\n", "EnderUNIX Aget v0.4");
66             usage();
67             exit(0);
68             break;
```

Find Help

Find Cancel

Path Report

Warning: Prev (M
at:

Begin by clicking

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Triaging Locksmith

Thread creation realizable?

The image shows a code editor window with a file named `main.c`. The code is as follows:

```
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc, argv, "p:l:n:hfv")) != -1) {
48     switch(c) {
49         case 'p':
50             req->port = atoi(optarg);
51             break;
52         case 'f':
53             fsuggested = 1;
54             break;
55         case 'l':
56             strncpy(req->lfile, optarg, MAXBUFSIZ);
57             break;
58         case 'n':
59             if ((nthreads = atoi(optarg)) > MAXTHREADS) {
60                 Log("Error: Maximum # of threads allowed is %
61                 nthreads = 0;
62             }
63             break;
64         case 'h':
65             printf("%s\n", "EnderUNIX Aget v0.4");
66             usage();
67             exit(0);
68             break;
```

The `pthread_create` function call on line 42 is highlighted in red. A pink box highlights the question "Thread creation realizable?".

The Path Report window on the right shows the following:

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

A pink arrow points to the `pthread_create()` call in the first path entry.

Triaging Locksmith

Thread creation realizable?

Yes, unconditionally created

```
main.c
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
    extern char *optarg;
    extern int optind;
    int c; int error = 0; int ret;
    struct hist_data h;
    int retlog;

    /* Allocate heap for download request
     * struct request stores all the information that might be
     * of interest
     */
    req = (struct request *)calloc(1, sizeof(struct request));

    /* Only some signals will be emitted */
    sigemptyset(&signal_set);
    sigaddset(&signal_set, SIGINT);
    sigaddset(&signal_set, SIGALRM);

    /* Block out all signals */
    pthread_sigmask(1, &signal_set, NULL);

    /* Create a thread for handling signals */
    if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0)
        fprintf(stderr, "main: cannot create signal_waiter thread: %s, error: %d", strerror(ret), ret);
    exit(-1);
}
```

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)
at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

Keep around context

The image shows a code editor window with a file named `main.c`. The code is a C program that demonstrates a data race. It includes headers for `errno`, `pthread`, `signal`, `unistd`, `stdio`, and `stdlib`. The `main` function sets up a signal handler and creates a thread to handle signals. A warning from Locksmith is shown in a `Path Report` window, indicating a possible data race of `prev` (Misc.c:<global>:15) at two locations in `main.c`. The first location is at `main():42` where `pthread_create()` is called. The second location is at `main():117` where `pthread_create()` is called. Both locations lead to a `dereference` in `updateProgressBar():193`. The `pthread_create` function is highlighted in red in the code editor.

```
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28      * struct request stores all the information that might be
29      * of interest
30      */
31     req = (struct request *)calloc(1, sizeof(struct request));
32
33     /* Only some signals will be emitted */
34     sigemptyset(&signal_set);
35     sigaddset(&signal_set, SIGINT);
36     sigaddset(&signal_set, SIGALRM);
37
38     /* Block out all signals */
39     pthread_sigmask(1, &signal_set, NULL);
40
41     /* Create a thread for hadling signals */
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43         fprintf(stderr, "main: cannot create signal_waiter thread: %s, e
44         exit(-1);
```

Path Report

Warning: Possible data race of `prev` (Misc.c:<global>:15) at:

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Triaging Locksmith

Keep around context

```
main.c 40
41 /* Create a thread for hadling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc, argv, "p:l:n:hfv")) != -1) {
48     switch(c) {
49
50
51
52     case 'f':
```

Easier than remembering

```
main.c 16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28     * struct request stores all the information that might be
```

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)
at:

1. <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

Path Visualizer

main.c split close

```
main.c
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc, argv, "p:l:n:hfv")) != -1) {
48     switch(c) {
49         case 'p':
50             req->port = atoi(optarg);
51             break;
52         case 'f':
```

main.c split close

```
main.c
16
17 extern int errno;
18
19 int main(int argc, char **argv)
20 {
21     extern char *optarg;
22     extern int optind;
23     int c; int error = 0; int ret;
24     struct hist_data h;
25     int retlog;
26
27     /* Allocate heap for download request
28     * struct request stores all the information that might be
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Focus

Triaging Locksmith

Path Visualizer

main.c

```
40
41 /* Create a thread for hadling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc, argv, "p:l:n:hfv")) != -1) {
48     switch(c) {
49         cas
50
51         break;
52     case 'f':
```

Call site of

signal_waiter, NULL)) != 0

Signal.c

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
22
23     arg = NULL;
24
25     pthread_sigmask(2, &signal_set, NULL);
26
27     while(1) {
28
29
```

Find

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

Next call -> pthread_create()

signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

Path Visualizer

main.c split close

```
40
41 /* Create a thread for hadling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0)
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, e
44     exit(-1);
45 }
46
47 while (!error && (c = getopt(argc,argv,"p:l:n:hfv")) != -1) {
```

Signal.c split close

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
22
23     arg = NULL;
24
```

Signal.c split close

```
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
60     printf("Signal Alarm came\n");
61     updateProgressBar(bwritten, req->clength);
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Skipping a few steps

Triaging Locksmith

Path Visualizer

main.c split close

```
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45
```

Signal.c split close

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
```

Signal.c split close

```
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
```

Misc.c split close

```
191     putchar(' ');
192     printf("[%d%% completed]\n", ndot);
193     prev = ndot;
194 }
195
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

locks: -

locks: -

the dereference, finally!

Triaging Locksmith

Path Visualizer

main.c split close

```
main.c
40
41 /* Create a thread for hadling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45
```

Signal.c split close

```
Signal.c
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
22
```

Signal.c split close

```
Signal.c
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
```

Misc.c split close

```
Misc.c
191     putchar(' ');
192     printf("[%d%% completed]\n", ndot);
193     prev = ndot;
194 }
195
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Triaging Locksmith

Path Visualizer

main.c split close

```
40
41 /* Create a thread for hadling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, e
44     exit(-1);
45
```

Signal.c split close

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
22
```

Signal.c split close

```
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
```

Misc.c split close

```
191     putchar(' ');
192     printf("[%d%% completed]\n", ndot);
193     prev = ndot;
194 }
195
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Screen is very cluttered!

Triaging Locksmith

Path Visualizer

main.c split close

```
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, ex
44     exit(-1);
45
```

Signal.c split close

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
22
```

Signal.c split close

```
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
```

Misc.c split close

```
191     putchar(' ');
192     printf("[%d%% completed]\n", ndot);
193     prev = ndot;
194 }
195
```

Find Help

Find Cancel

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Old context is hidden!

Triaging Locksmith

The screenshot shows the Path Visualizer application with four code snippets and a Path Report panel on the right. The snippets are for main.c, Signal.c, Signal.c, and Misc.c. The Path Report panel shows a warning about a possible data race and lists two call paths.

main.c (lines 40-45):

```
40
41 /* Create a thread for handling signals */
42 if ((ret = pthread_create(&hthread, NULL, signal_waiter, NULL)) != 0
43     fprintf(stderr, "main: cannot create signal_waiter thread: %s, e
44     exit(-1);
45
```

Signal.c (lines 17-22):

```
17 extern pthread_mutex_t bwritten_mutex;
18
19 void * signal_waiter(void *arg)
20 {
21     int signal;
```

Signal.c (lines 54-59):

```
54     exit(0);
55 }
56
57
58 void sigalrm_handler(void)
59 {
```

Misc.c (lines 191-195):

```
191     putchar(' ');
192     printf("[%d%% completed]\n", ndot);
193     prev = ndot;
194 }
195
```

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Which function is this?

Old context is hidden!

Where was this called?

A Thousand Cuts

A Thousand Cuts

- Read error report
- Click hyperlink 1
- Read code
- Scroll up
- Scroll down
- Split window
- Focus
- Back to error report
- Click hyperlink 2
- Read code
- Scroll down
- Split window
- Focus
- Back to error report
- Click hyperlink 3
- Read code
- Scroll down
- Split window
- Focus
- Back to error report
- Collapse splits
- (resize window, move window...)

• Many little distractions from actual task

• Seemingly straightforward task becomes complex!

Path Projection

- Designed for tracing paths

```
main.c - main()
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter, arg)))
        return -1;
}

main.c - signal_waiter()
19 void * signal_waiter(void *arg)
20 {
27     while(1) {
33         if (signal == SIGINT) {
35             } else if (signal == SIGALRM) {
36                 sigalrm_handler();
}

Signal.c - sigalrm_handler()
58 void sigalrm_handler(void)
59 {
61     updateProgressBar(bwritten, req->clength);
}

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
63 }
37 }
38 }
39 }
45 }
122 }
```

Path Projection

- Designed for tracing paths
- Function call inlining:
Inline function directly below call site

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter, arg)))
        return ret;
}

main.c - signal_waiter()
19 void * signal_waiter(void *arg)
20 {
27     while(1) {
33         if (signal == SIGINT) {
35             } else if (signal == SIGALRM) {
36                 sigalrm_handler();
}

Signal.c - sigalrm_handler()
58 void sigalrm_handler(void)
59 {
61     updateProgressBar(bwritten, req->clength);
}

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
63 }
37 }
38 }
39 }
45 }
122 }
```

Path Projection

- Designed for tracing paths
- Function call inlining:
Inline function directly below call site
- Path-derived code folding:
Show only implicated lines and lexical control-blocks

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter, arg)))
        return -1;
}

19 void * signal_waiter(void *arg)
20 {
27     while(1){
33         if (signal == SIGINT) {
35             } else if (signal == SIGALRM) {
36                 signalrm_handler();
}

Signal.c - signalrm_handler()
58 void signalrm_handler(void)
59 {
61     updateProgressBar(bwritten, req->clength);
}

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
63 }
37 }
38 }
39 }
45 }
122 }
```

Path Projection

- Designed for tracing paths
- Function call inlining:
 - Inline function directly below call site
- Path-derived code folding:
 - Show only implicated lines and lexical control-blocks
- Show as much code as possible on one screen

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter, &req)))
    return ret;

    while(1) {
        if (signal == SIGINT) {
        } else if (signal == SIGALRM) {
            sigalrm_handler();
        }
    }
}

void * signal_waiter(void *arg)
{
    while(1) {
        if (signal == SIGINT) {
        } else if (signal == SIGALRM) {
            sigalrm_handler();
        }
    }
}

void sigalrm_handler(void)
{
    updateProgressBar(bwritten, req->clength);
}

void updateProgressBar(float cur, float tot)
{
    prev = ndot;
}

122 }
```

main.c - main()

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter,
```

Signal.c - signal_waiter()

```
19 void * signal_waiter(void *arg)
20 {
27     while(1){
33         if (signal == SIGINT){
35             } else if (signal == SIGALRM){
36                 sigalrm_handler();
```

Signal.c - sigalrm_handler()

```
58 void sigalrm_handler(void)
59 {
61     updateProgressBar(bwritten, req->clength);
```

Misc.c - updateProgressBar()

```
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
```

```
63 }
37     }
38 }
39 }
```

```
45 }
122 }
```

Path Projection

Path Visualizer

Path 1 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter,
Signal.c - signal_waiter()
19 void * signal_waiter(void *arg)
20 {
27     while(1){
33         if (signal == SIGINT){
35         } else if (signal == SIGALRM){
36             sigalrm_handler();
Signal.c - sigalrm_handler()
58 void sigalrm_handler(void)
59 {
61     updateProgressBar(bwritten, req->clength);
Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
63 }
37 }
38 }
39 }
45 }
122 }
```

main.c - main() 19 in
20 {
42 {
45
116
117

main.c - main() 133
134
166
171

Aget.c - resume_get() 172
174
175
177

Download.c - http_get() 172
174
175
177

Multi-query

Find

- ✗ "pthread_create"
- ✗ "pthread_join"
- ✗ "pthread_mutex_lock"
- ✗ "pthread_mutex_unlock"

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Show paths side by side

Path Projection

The screenshot shows the Path Visualizer tool interface. The main window displays a code path for 'Path 2 of 2'. The code is color-coded and includes comments for file and line numbers. The path starts in `main.c` at line 19, goes to `pthread_create` at line 42, then to `read_log` at line 116, and `resume_get` at line 117. From `resume_get`, it goes to `http_get` at line 37, then to `updateProgressBar` at line 121. The `updateProgressBar` function is shown in a separate window, with lines 177-194. The path ends with a `dereference` warning at line 193.

The right-hand pane shows a 'Multi-query' section with a search bar and a list of queries: `"pthread_create"`, `"pthread_join"`, `"pthread_mutex_lock"`, and `"pthread_mutex_unlock"`. Below this is a 'Path Report' section with a warning: 'Warning: Possible data race of prev (Misc.c:<global>:15) at:'. The report lists two paths:

- `<in main.c>`
`main():42` -> `pthread_create()`
`<in Signal.c>`
`signal_waiter():36`
`sigarm_handler():61`
`<in Misc.c>`
`updateProgressBar():193` -> dereference
locks: -
- `<in main.c>`
`main():117`
`<in Aget.c>`
`resume_get():171` -> `pthread_create()`
`<in Download.c>`
`http_get():121`
`<in Misc.c>`
`updateProgressBar():193` -> dereference
locks: -

A pink callout box at the bottom left contains the text 'Show paths side by side'.

Path Projection

Path Visualizer

Path 2 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mute
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }

172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

Multi-query

Find

- ✗ "pthread_create"
- ✗ "pthread_join"
- ✗ "pthread_mutex_lock"
- ✗ "pthread_mutex_unlock"

Multiple searches (despite folds)

main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Path Projection

The image shows a screenshot of the Path Visualizer application. The main window displays a code path titled "Path 2 of 2". The code is color-coded and includes several pthread-related functions. A search panel on the right, titled "Multi-query", lists four search terms: "pthread_create", "pthread_join", "pthread_mutex_lock", and "pthread_mutex_unlock". A callout box with a pink border contains the text "Multiple searches (despite folds)". Pink arrows point from this box to the search terms in the Multi-query panel and to the corresponding function calls in the code. The code path includes:

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&wthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mutex);
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }

172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

The Multi-query panel shows the following search results:

```
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

2. <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
```

Path Projection

Continuing example...

```
main.c - main()
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

Aget.c - resume_get()
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

Download.c - http_get()
37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex);
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mutex);
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }

172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

```
Find
x "pthread_create"
x "pthread_join"
x "pthread_mutex_lock"
x "pthread_mutex_unlock"
```

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)
at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Path Projection

from 1st
call stack

```
Path 2 of 2
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_waiter,
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_get,
37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex);
121         updateProgressBar(bwritten, td->clength);
Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mutex);
127     pthread_mutex_unlock(&http_get_ok_mutex);
128 }
135 }
172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

Multi-query

Help

Find

```
x "pthread_create"
x "pthread_join"
x "pthread_mutex_lock"
x "pthread_mutex_unlock"
```

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)
at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Path Projection

Path Visualizer

Path 2 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mute
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }

172 }
174     for (i = 0; i < nthreads; i++)
175         pthread_join(wthread[i].tid, NULL);
177     if (http_get_ok == nthreads)
```

from 2nd call stack

Multi-query

- pthread_create
- pthread_join
- pthread_mutex_lock
- pthread_mutex_unlock

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Path Projection

Path Visualizer

Path 2 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_
37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)
Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mute
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }
172 }
174     for (i = 0; i < nthreads; i++)
175         pthread_join(wthread[i].tid, NULL);
177     if (http_get_ok == nthreads)
```

from 2nd call stack

dereference

Multi-query

Find

- ✗ "pthread_create"
- ✗ "pthread_join"
- ✗ "pthread_mutex_lock"
- ✗ "pthread_mutex_unlock"

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Path Projection

Path Visualizer

Path 2 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_
37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mute
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }
172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

condition for thread creation

in loop

condition for dereference

Multi-query

Find

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigalrm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Path Projection

Path Visualizer

Path 2 of 2

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex)
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124     if (td->offset == td->foffset) {
125         pthread_mutex_lock(&http_get_ok_mute
127         pthread_mutex_unlock(&http_get_ok_mu

174     for (i = 0; i < nthreads; i++)
175         pthread_join(wthread[i].tid, NULL);
177     if (http_get_ok == nthreads)
```

Multi-query

Find

- ✗ "pthread_create"
- ✗ "pthread_join"
- ✗ "pthread_mutex_lock"
- ✗ "pthread_mutex_unlock"

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

not a single click or scroll!

Path Projection

The screenshot shows the Path Visualizer tool interface. The main window displays a call stack with the following entries:

- main.c - main():
 - 19 int main(int argc, char **argv)
 - 20 {
 - 42 if ((ret = pthread_create(&hthread, NULL, signal_wait
 - 45 }
 - 116 if ((retlog = read_log(&h)) != -1)
 - 117 resume_get(&h);
- Aget.c - resume_get():
 - 133 void resume_get(struct hist_data *h)
 - 134 {
 - 166 for (i = 0; i < nthreads; i++) {
 - 171 pthread_create(&(wthread[i].tid), NULL, http_
- Download.c - http_get():
 - 37 void * http_get(void *arg) {
 - 106 pthread_mutex_lock(&bwritten_mutex);
 - 108 pthread_mutex_unlock(&bwritten_mutex);
 - 110 while (td->offset < foffset) {
 - 118 pthread_mutex_lock(&bwritten_mutex);
 - 120 pthread_mutex_unlock(&bwritten_mutex)
 - 121 updateProgressBar(bwritten, td->clength)
- Misc.c - updateProgressBar():
 - 177 void updateProgressBar(float cur, float tot)
 - 178 {
 - 193 prev = ndot;
 - 194 }
- Download.c - http_get():
 - 122 }
 - 124 if (td->offset == td->foffset) {
 - 125 pthread_mutex_lock(&http_get_ok_mute
 - 127 pthread_mutex_unlock(&http_get_ok_mu
- main.c - main():
 - 174 for (i = 0; i < nthreads; i++)
 - 175 pthread_join(wthread[i].tid, NULL);
 - 177 if (http_get_ok == nthreads)

The right-hand pane shows the Multi-query and Path Report sections. The Multi-query section contains a list of pthread functions:

- pthread_create
- pthread_join
- pthread_mutex_lock
- pthread_mutex_unlock

The Path Report section contains a warning and a call stack:

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
 - main():42 -> pthread_create()
 - <in Signal.c>
 - signal_waiter():36
 - sigarm_handler():61
 - <in Misc.c>
 - updateProgressBar():193 -> dereference
- <in main.c>
 - main():117
 - <in Aget.c>
 - resume_get():171 -> pthread_create()
 - <in Download.c>
 - http_
 - <in
 - up

Two callouts in pink boxes are present:

- not a single click or scroll!
- no need to look here too!

Path Projection

What's offset?

```
main.c - main()
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

Aget.c - resume_get()
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

Download.c - http_get()
37 void * http_get(void *arg) {
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
118         pthread_mutex_lock(&bwritten_mutex);
120         pthread_mutex_unlock(&bwritten_mutex);
121         updateProgressBar(bwritten, td->clength)

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }

122 }
124 if (td->offset == td->foffset) {
125     pthread_mutex_lock(&http_get_ok_mutex);
127     pthread_mutex_unlock(&http_get_ok_mu
128 }
135 }

172 }
174 for (i = 0; i < nthreads; i++)
175     pthread_join(wthread[i].tid, NULL);
177 if (http_get_ok == nthreads)
```

Multi-query

```
Find
x "pthread_create"
x "pthread_join"
x "pthread_mutex_lock"
x "pthread_mutex_unlock"
```

Path Report

Warning: Possible data race of
prev (Misc.c:<global>:15)
at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference

locks: -

Path Projection

Path Visualizer

Path 2 of 2

```
main.c - main()
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

Aget.c - resume_get()
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

Download.c - http_get()
37 void * http_get(void *arg) {
42     long foffset;
54     foffset = td->foffset;
99     if ((dr - i) > foffset)
100         dw = pwrite(td->fd, s, (foffset - i), td->soffset);
101     else
102         dw = pwrite(td->fd, s, (dr - i), td->soffset);
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
113         if ((td->offset + dr) > foffset)
114             dw = pwrite(td->fd, rbuf, foffset - td->offset, td->soffset);
115         else
116             dw = pwrite(td->fd, rbuf, dr, td->soffset);
118     pthread_mutex_lock(&bwritten_mutex);
120     pthread_mutex_unlock(&bwritten_mutex);
121     updateProgressBar(bwritten, td->clength);

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
```

Multi-query

foffset Find

- pthread_create
- pthread_join
- pthread_mutex_lock
- pthread_mutex_unlock
- foffset

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Path Projection

What's in read_log?

```
main.c - main()
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     })
116     if ((retlog = read_log(&h)) != -1)
117         resume_get(&h);

Aget.c - resume_get()
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_

Download.c - http_get()
37 void * http_get(void *arg) {
42     long foffset;
54     foffset = td->foffset;
99     if ((dr - i) > foffset)
100         dw = pwrite(td->fd, s, (foffset - i), td->soffset);
101     else
102         dw = pwrite(td->fd, s, (dr - i), td->soffset);
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
113         if ((td->offset + dr) > foffset)
114             dw = pwrite(td->fd, rbuf, foffset - td->offset, td->soffset);
115         else
116             dw = pwrite(td->fd, rbuf, dr, td->soffset);
118     pthread_mutex_lock(&bwritten_mutex);
120     pthread_mutex_unlock(&bwritten_mutex);
121     updateProgressBar(bwritten, td->clength);

Misc.c - updateProgressBar()
177 void updateProgressBar(float cur, float tot)
178 {
193     prev = ndot;
194 }
```

Multi-query

```
foffset Find
x "pthread_create"
x "pthread_join"
x "pthread_mutex_lock"
x "pthread_mutex_unlock"
x "foffset"
```

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) at:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Path Projection

The screenshot shows the Path Visualizer application. The main window displays a call stack path for 'Path 2 of 2'. The path starts in `main.c - main()` at line 117, where `read_log(&h)` is called. This leads to `Resume.c - read_log()` at line 52, which then calls `resume_get(&h)` at line 171. This function calls `pthread_create()`, which leads to `Aget.c - resume_get()` at line 37, then `Download.c - http_get()` at line 121, and finally `Misc.c - updateProgressBar()` at line 193. A pink callout box points to the `read_log()` call in the main.c stack frame, containing the text 'Reveal definition (initially folded)'. On the right, the 'Multi-query' panel shows a search for 'foffset' with several pthread-related functions listed. The 'Path Report' panel shows a warning about a possible data race and lists the call stack path with dereference points.

```
19 int main(int argc, char **argv)
20 {
42     if ((ret = pthread_create(&hthread, NULL, signal_wait
45     }
116    if ((retlog = read_log(&h)) != -1)
+Resume.c - read_log() x
52    int read_log(struct hist_data *h)
53    {
90    }
117    resume_get(&h)
+Aget.c - resume_get()
133 void resume_get(struct hist_data *h)
134 {
166     for (i = 0; i < nthreads; i++) {
171         pthread_create(&(wthread[i].tid), NULL, http_
+Download.c - http_get()
37 void * http_get(void *arg) {
42     long foffset;
54     foffset = td->foffset;
99     if ((dr - i) > foffset)
100         dw = pwrite(td->fd, s, (foffset - i), td->soffset);
101     else
102         dw = pwrite(td->fd, s, (dr - i), td->soffset);
106     pthread_mutex_lock(&bwritten_mutex);
108     pthread_mutex_unlock(&bwritten_mutex);
110     while (td->offset < foffset) {
113         if ((td->offset + dr) > foffset)
114             dw = pwrite(td->fd, rbuf, foffset - td->offset, td->soffset);
115         else
116             dw = pwrite(td->fd, rbuf, dr, td->offset);
118     pthread_mutex_lock(&bwritten_mutex);
120     pthread_mutex_unlock(&bwritten_mutex);
121     updateProgressBar(bwritten, td->clength);
+Misc.c - updateProgressBar()
193 void updateProgressBar(struct hist_data *h, int written)
194 {
195     if (written > 0)
196         progress_bar->value += written;
197     if (progress_bar->value > progress_bar->total)
198         progress_bar->value = progress_bar->total;
199     if (progress_bar->value > 0)
200         progress_bar->percent = (progress_bar->value * 100) / progress_bar->total;
201     if (progress_bar->percent > 100)
202         progress_bar->percent = 100;
203     if (progress_bar->percent < 0)
204         progress_bar->percent = 0;
205     if (progress_bar->percent > 0)
206         progress_bar->label = sprintf(progress_bar->label, "%d%%", progress_bar->percent);
207     if (progress_bar->label[0] != '\0')
208         printf("%s\r\n", progress_bar->label);
209 }
```

Multi-query
foffset Find

- x "pthread_create"
- x "pthread_join"
- x "pthread_mutex_lock"
- x "pthread_mutex_unlock"
- x "foffset"

Path Report

Warning: Possible data race of prev (Misc.c:<global>:15) t:

- <in main.c>
main():42 -> pthread_create()
<in Signal.c>
signal_waiter():36
sigarm_handler():61
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -
- <in main.c>
main():117
<in Aget.c>
resume_get():171 -> pthread_create()
<in Download.c>
http_get():121
<in Misc.c>
updateProgressBar():193 -> dereference
locks: -

Pilot User Study

- We discovered that static analysis is...

Rocket Science

- In our pilot studies, non-expert users had great trouble triaging Locksmith error reports:
 - ad hoc, inconsistent procedure
 - neglected some causes of false positives
 - sidetracked by non-causes of false positives
- Even with extensive tutorials!

Rocket Science 101

- Our solution: **triaging checklist**
- Checklists are **tool-/error-specific**
 - Different tools have different imprecision & error reports
- Anecdotally, **41%** faster at triaging using checklist

Locksmith Triaging Checklist

- To triage Locksmith:

check if any pair of paths are simultaneously realizable

different cases: threads in loop, parent-child, child-child

- For example:

Source of imprecision: Locksmith is path-insensitive

Possible false positive: child-child threads may be mutually exclusive

For threads leading to dereferences in Paths 1 and 2:

Are they parent-child (or child-parent), or child-child?

Parent-child / Child-child

Child-child threads.

Y N

Are the children mutually exclusive (i.e., **only one** can be spawned by their common parent/ancestor)?

If no, there is likely a race. Are there reasons to show otherwise?

????

Explain: ????

Locksmith Triaging Checklist

- To triage Locksmith:

check if any pair of paths are simultaneously realizable

different cases: threads in loop, parent-child, child-child

- For example:

Source of imprecision: Locksmith is path-insensitive

Possible false positive: child-child threads may be mutually exclusive

For threads leading to dereferences in Paths 1 and 2:

Are they parent-child (or child-parent), or child-child?

Parent-child / Child-child

Child-child threads.

	Y	N
Are the children mutually exclusive (i.e., only one can be spawned by their common parent/ancestor)?	<input type="radio"/>	<input type="radio"/>
If no, there is likely a race. Are there reasons to show otherwise?		????
Explain: ????		

Locksmith Triaging Checklist

- To triage Locksmith:

check if any pair of paths are simultaneously realizable

different cases: threads in loop, parent-child, child-child

- For example:

Source of imprecision: Locksmith is path-insensitive

Possible false positive: child-child threads may be mutually exclusive

For threads leading to dereferences in Paths 1 and 2:

Are they parent-child (or child-parent), or child-child?

Parent-child / Child-child

Child-child threads.

	Y	N
Are the children mutually exclusive (i.e., only one can be spawned by their common parent/ancestor)?	<input type="radio"/>	<input type="radio"/>
If no, there is likely a race. Are there reasons to show otherwise?		????
Explain: ????		

User Study

- Which is better: Standard Viewer (SV) or Path Projection (PP)?
 - Quantitatively: completion time
 - Qualitatively: user ratings
- Data race triaging task using Locksmith

User Study Issues

- Large variance between participants
 - Participants have different skill level
 - Are differences due to participant or UI?
- Within-subjects: each participant use both interfaces
 - Compare UI results for each participant

User Study Issues

- Order and carryover effect
 - Participants get better over time (learning)
 - Participants biased by initial UI or problem
- Counter-balance: divide participants into two groups
 - SV-PP: Standard Viewer, then Path Projection
 - PP-SV: Path Projection, then Standard Viewer

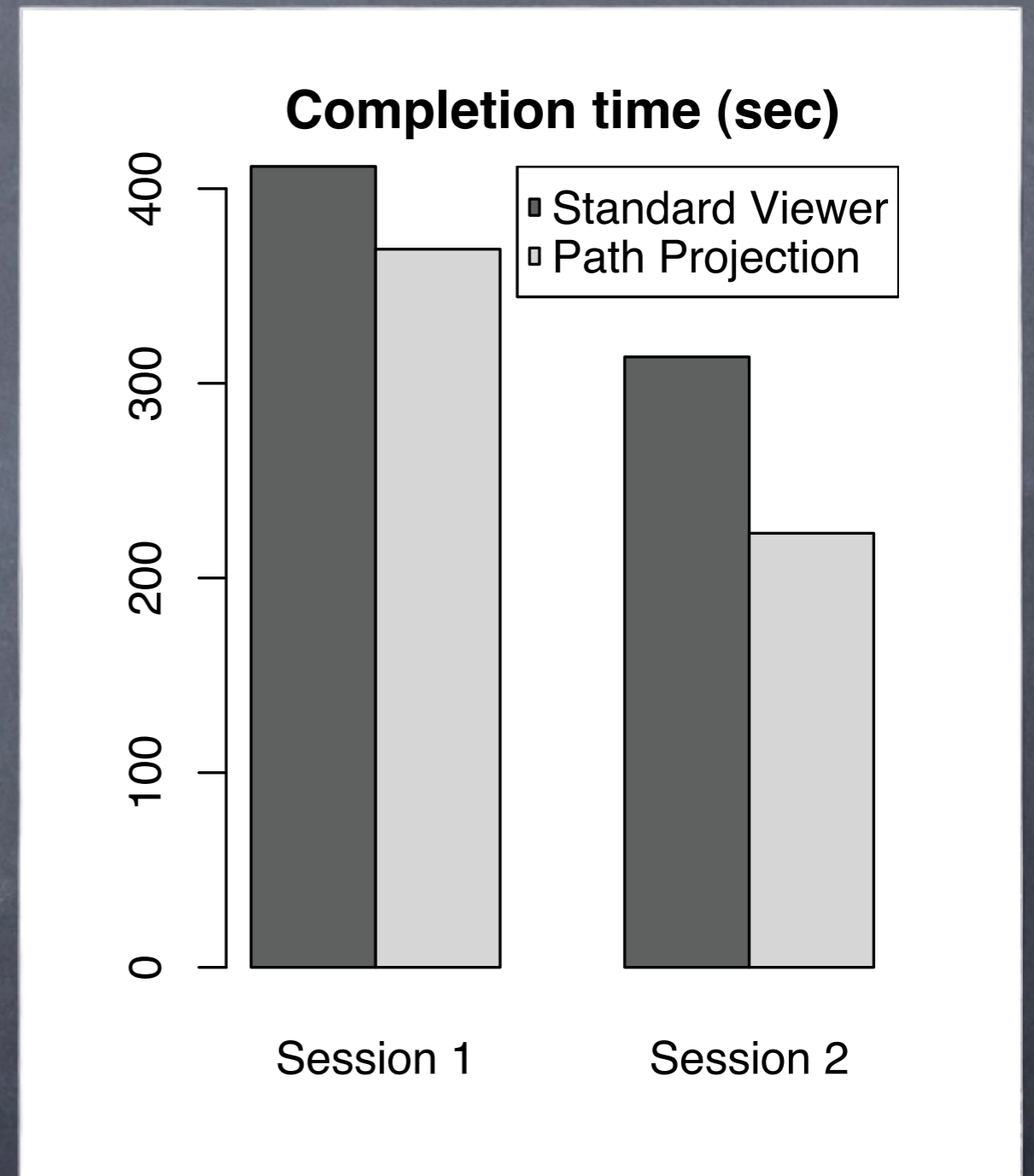
User Study: Locksmith Task

- 6 trials from Locksmith corpus (unfamiliar to users)
- One warning per trial
 - no need to manage warnings
- Only verify that paths are simultaneously realizable
 - No aliasing/imprecise lock state (future work)

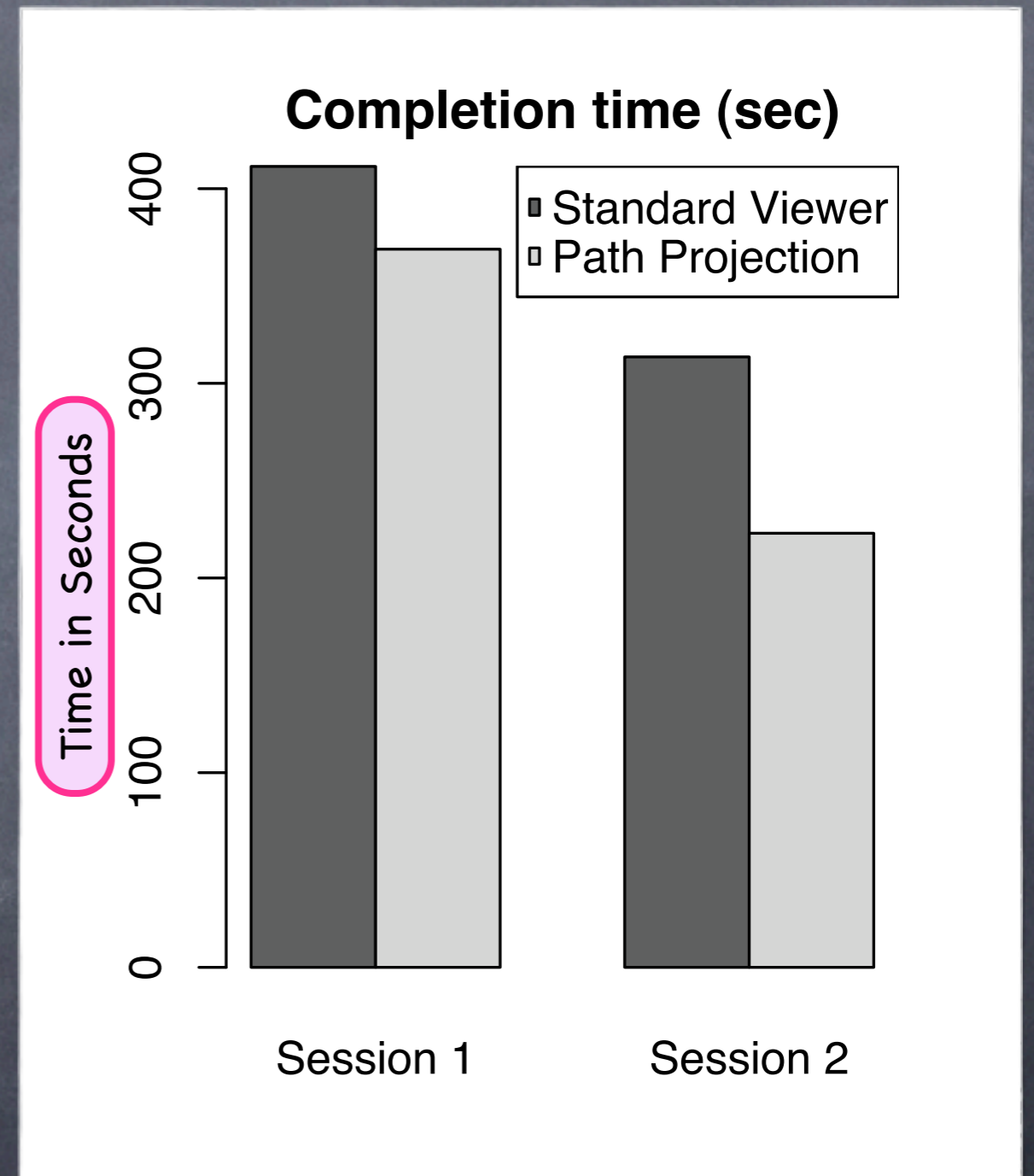
User Study: Misc.

- 8 student participants
 - 3 undergraduates, 5 graduates
 - Prior experience in C, multithreading (not necessarily C)
 - Self-rated 3-4 (1: no experience to 5: very experienced)
 - 2 had experience in Locksmith and Eraser

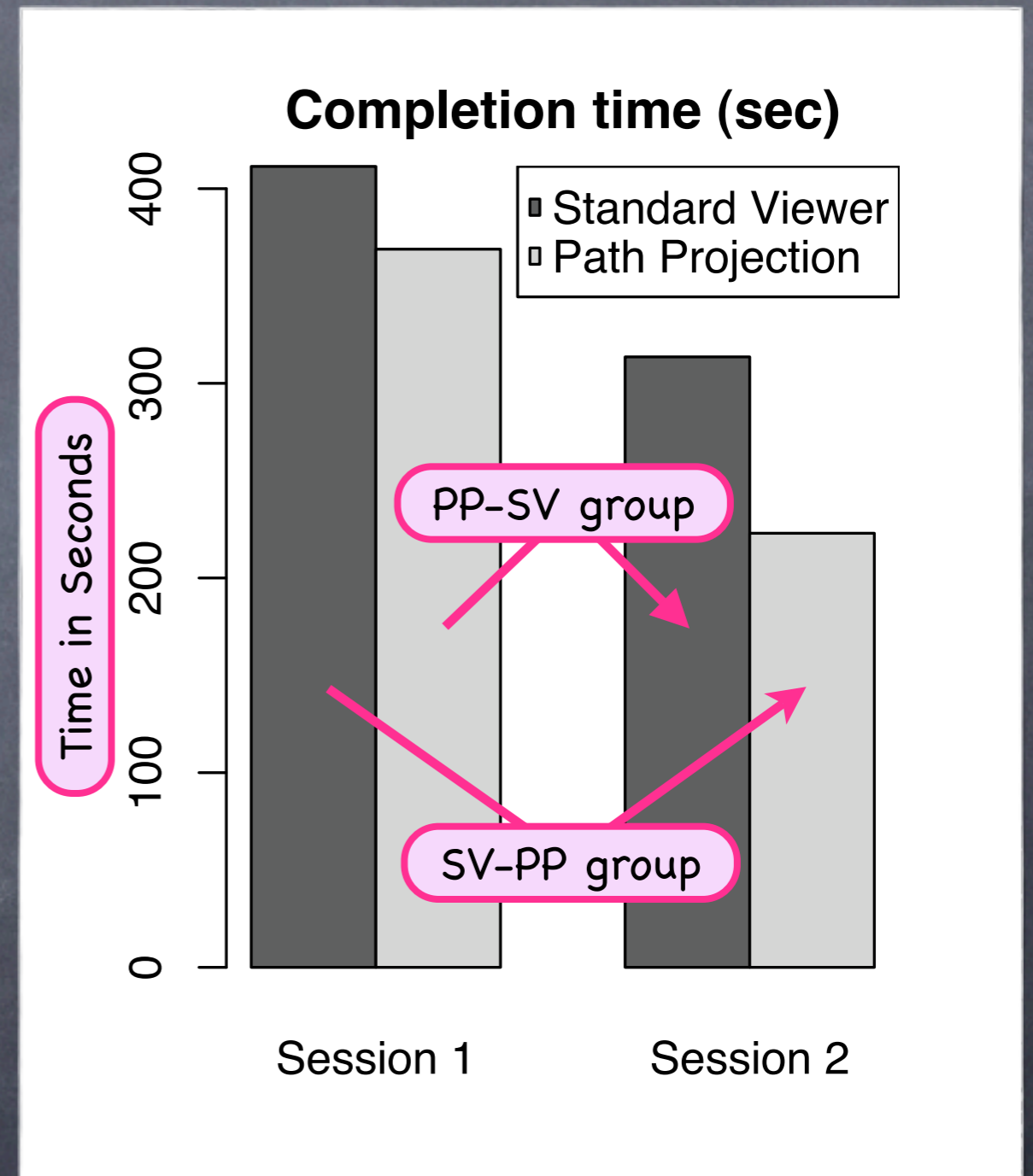
Quantitative (Chart guide)



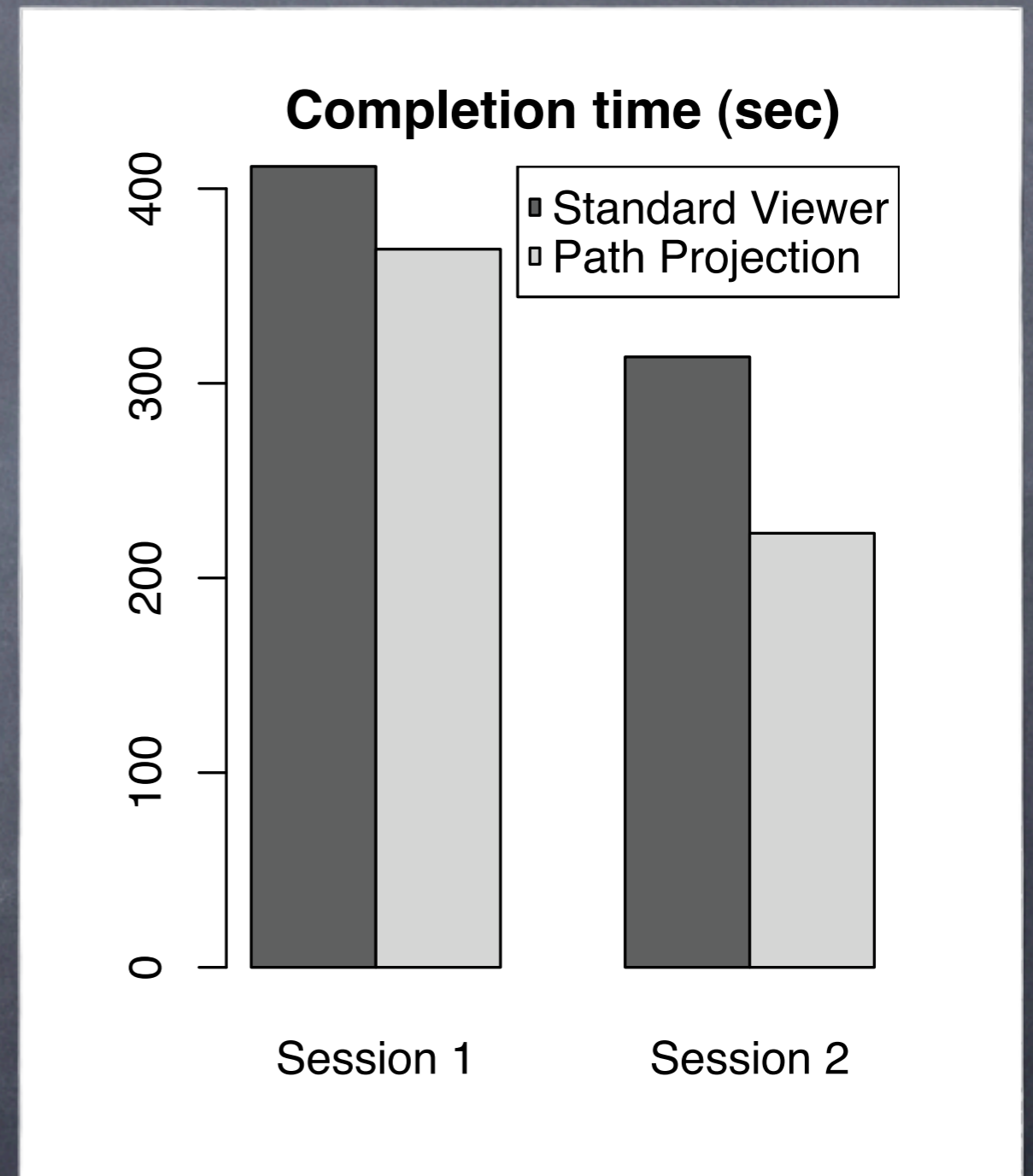
Quantitative (Chart guide)



Quantitative (Chart guide)



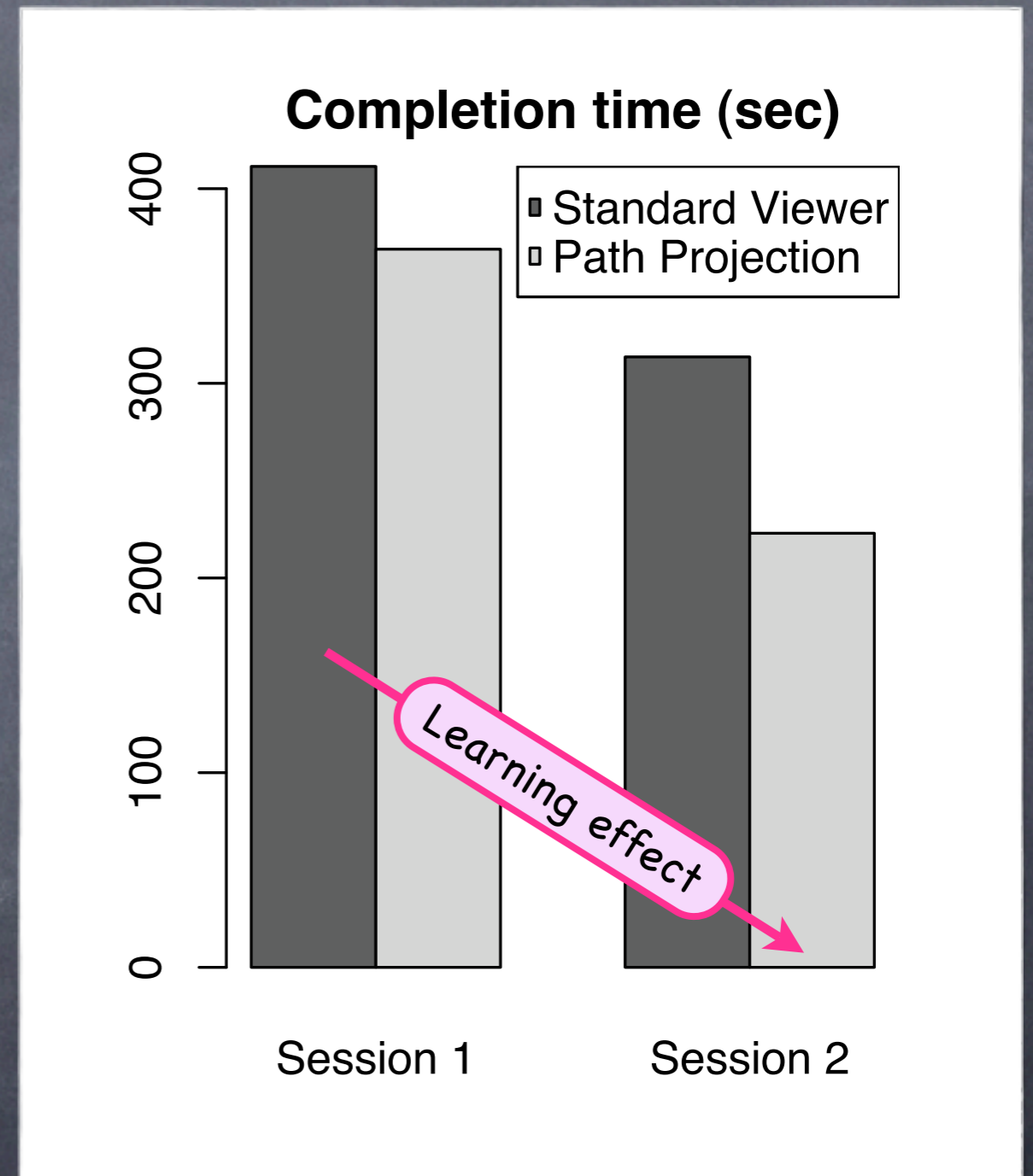
Faster Completion Time



Faster Completion Time

- Learning effect

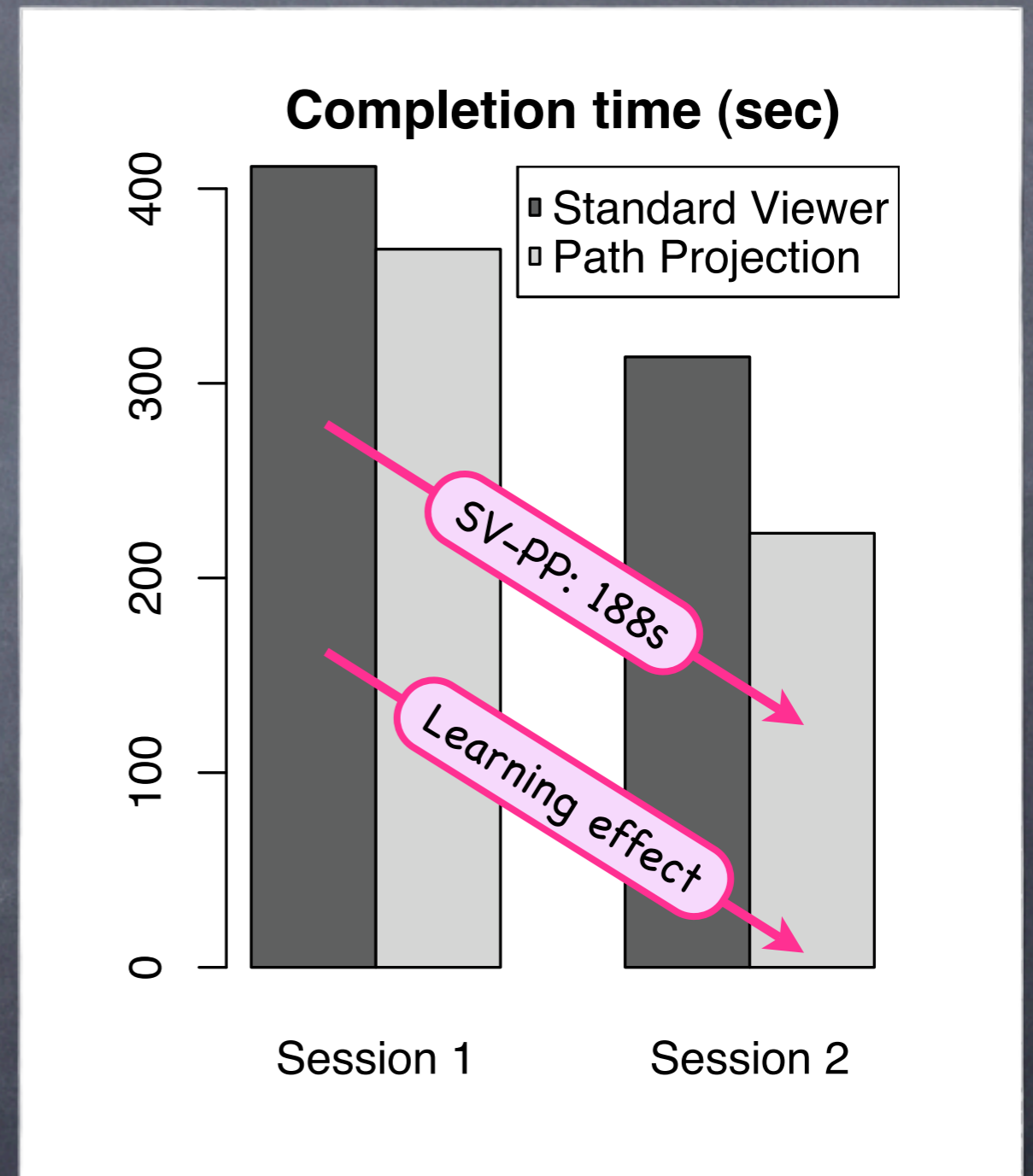
all improved in Session 2*



*statistically significant ($p < 0.05$)

Faster Completion Time

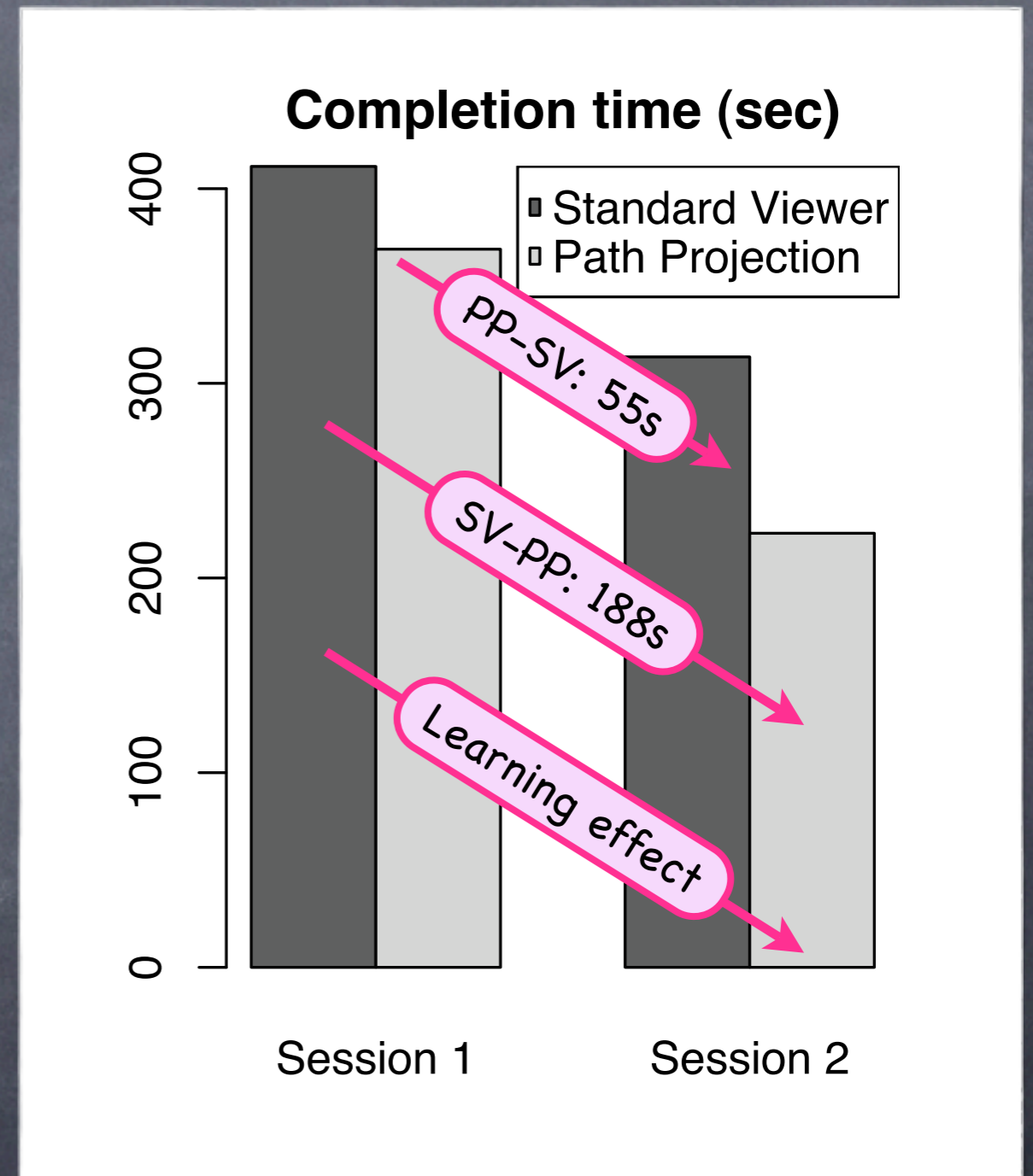
- Learning effect
all improved in Session 2*
- SV-PP improved by 188s*
(effect size $d=1.276$)



*statistically significant ($p<0.05$)

Faster Completion Time

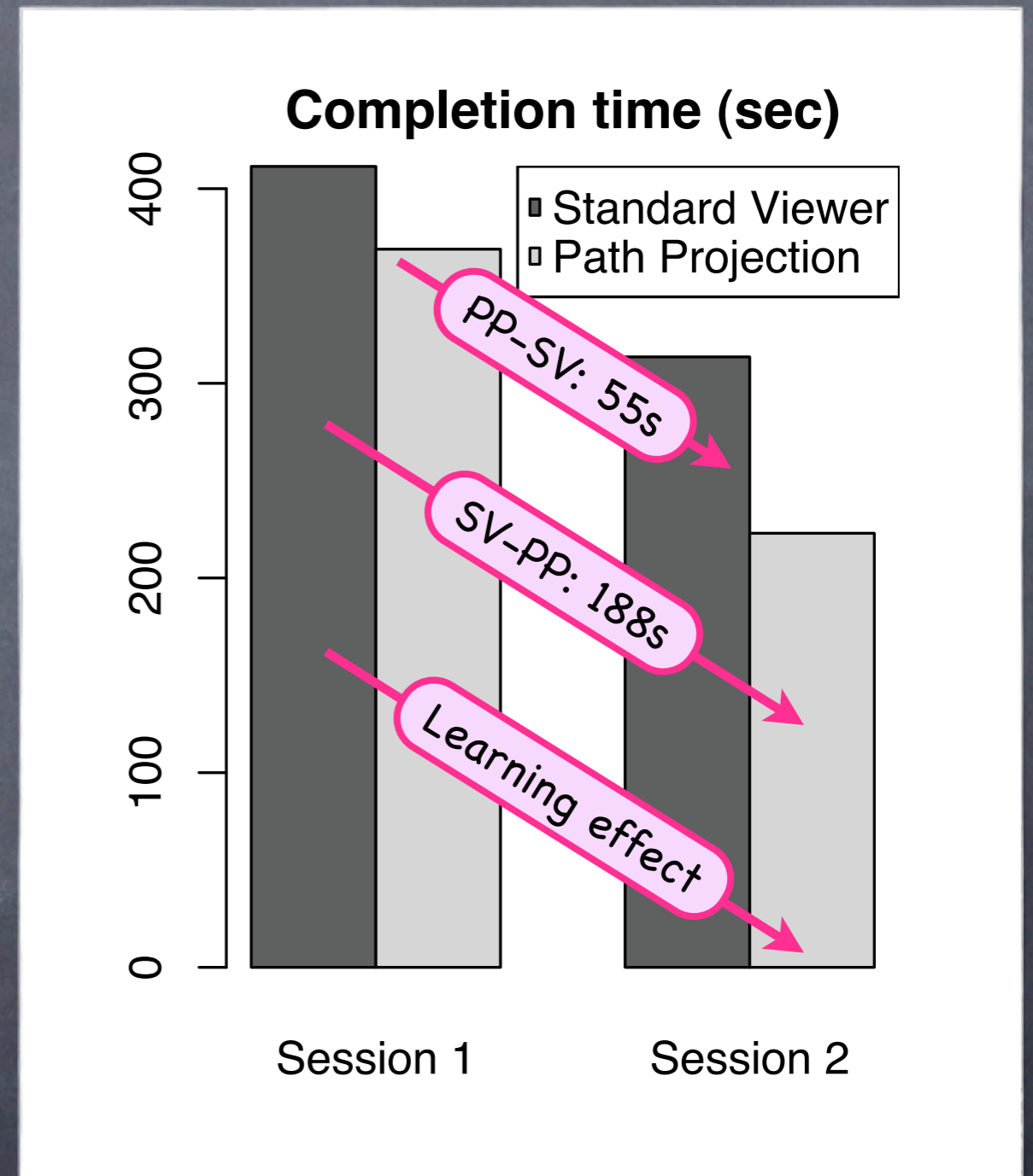
- Learning effect
all improved in Session 2*
- SV-PP improved by 188s*
(effect size $d=1.276$)
- PP-SV improved by 55s*
(effect size $d=0.375$)



*statistically significant ($p<0.05$)

Faster Completion Time

- Learning effect
 - all improved in Session 2*
- SV-PP improved by 188s*
(effect size $d=1.276$)
- PP-SV improved by 55s*
(effect size $d=0.375$)
- Similar # mistakes
10 in PP (10.9%), 9 in SV (9.8%)

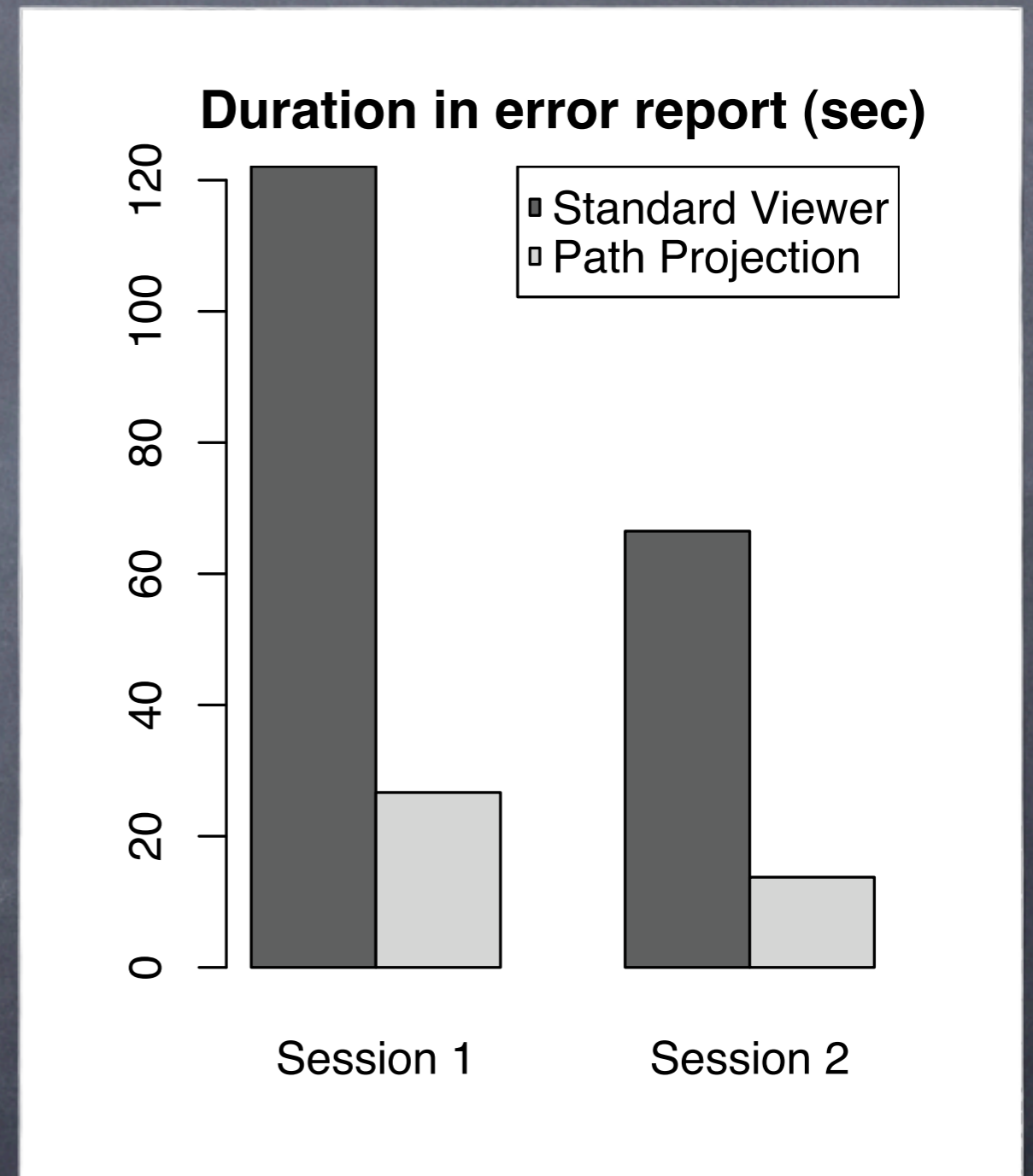


- **18%** faster on average

*statistically significant ($p<0.05$)

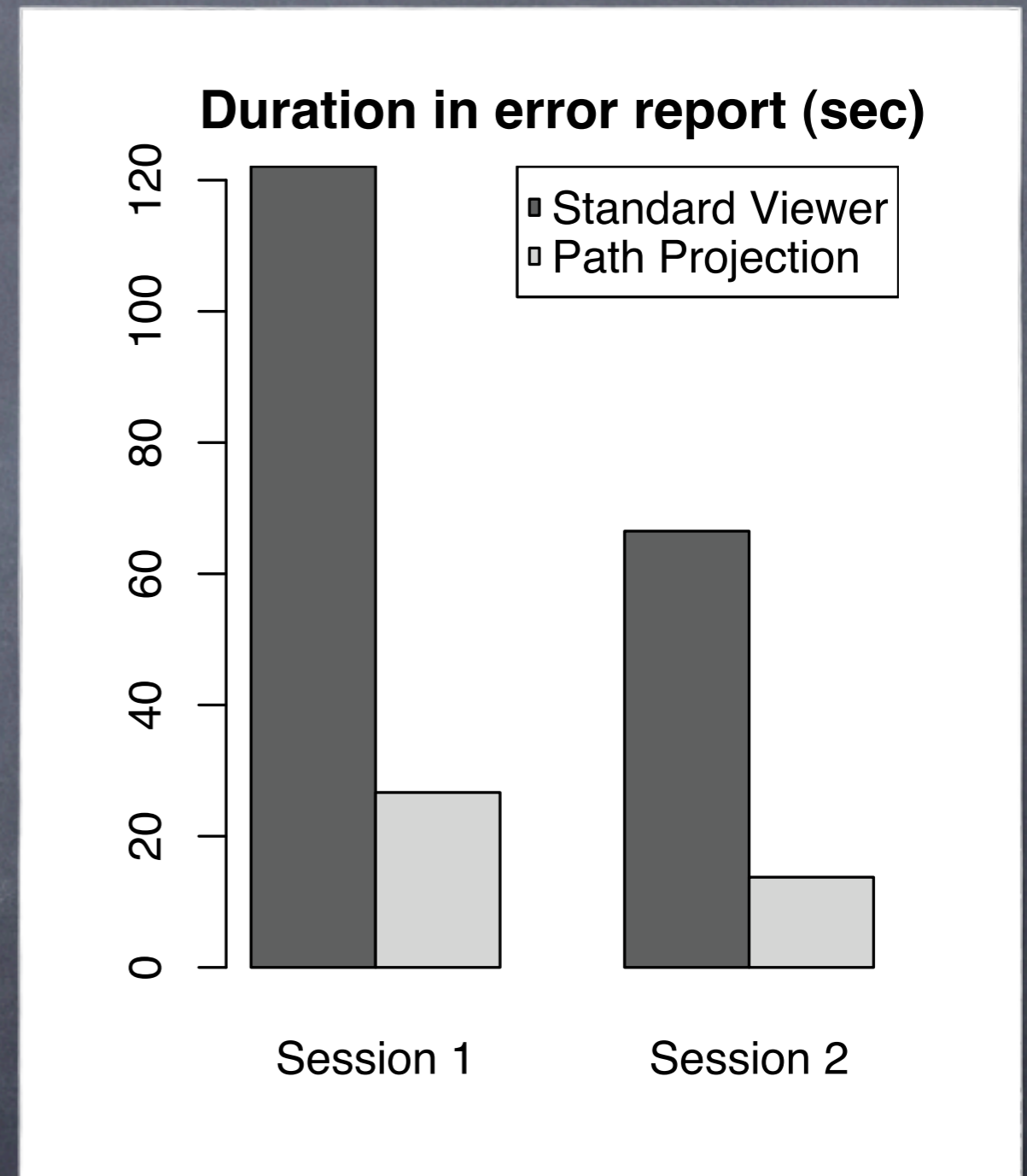
Less Use of Error Report

- Duration where pointer is over error report
(e.g., using hyperlinks)



Less Use of Error Report

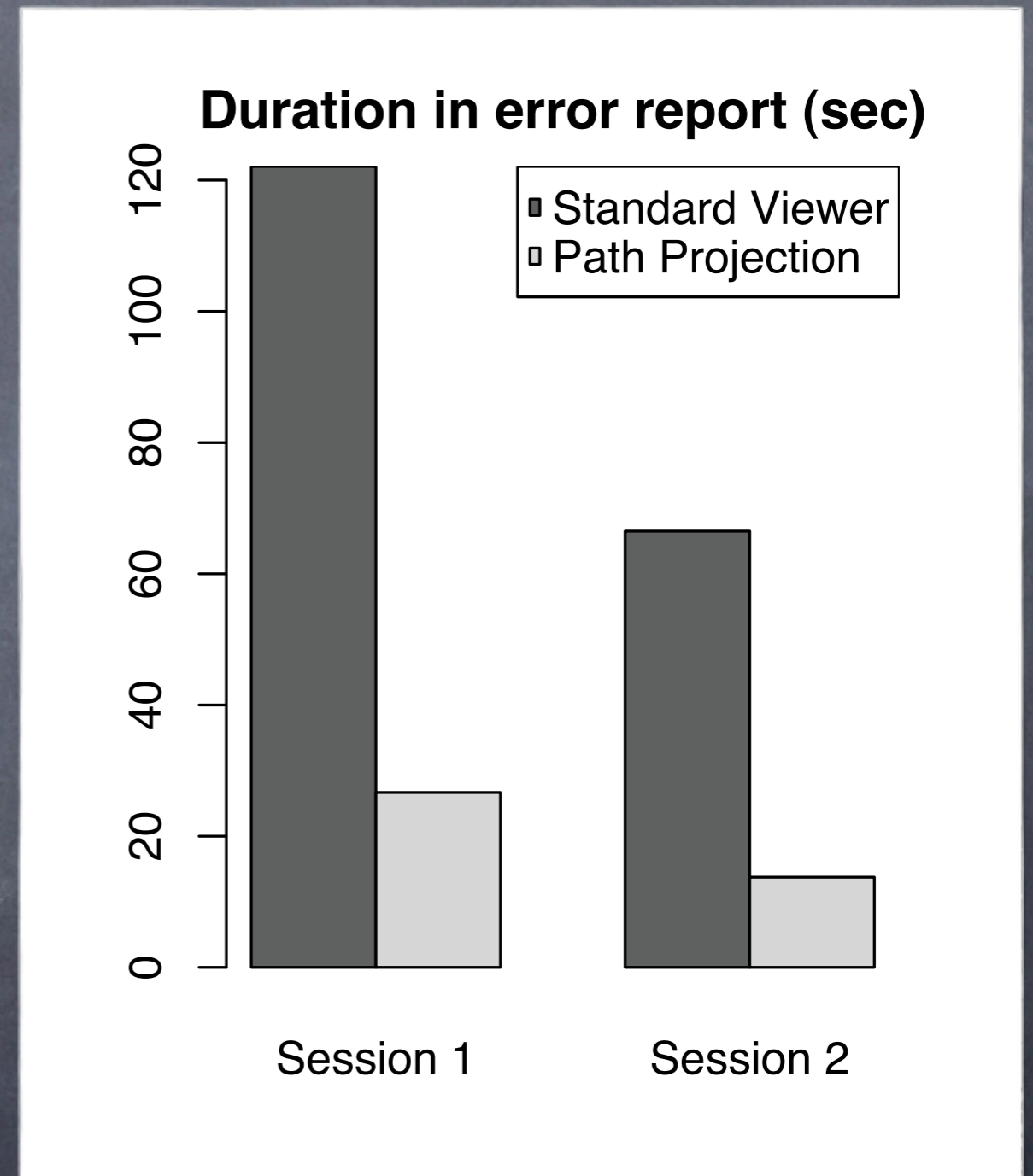
- Duration where pointer is over error report
(e.g., using hyperlinks)
- On average, only 20s with PP vs. 94s with SV*



*statistically significant ($p < 0.05$)

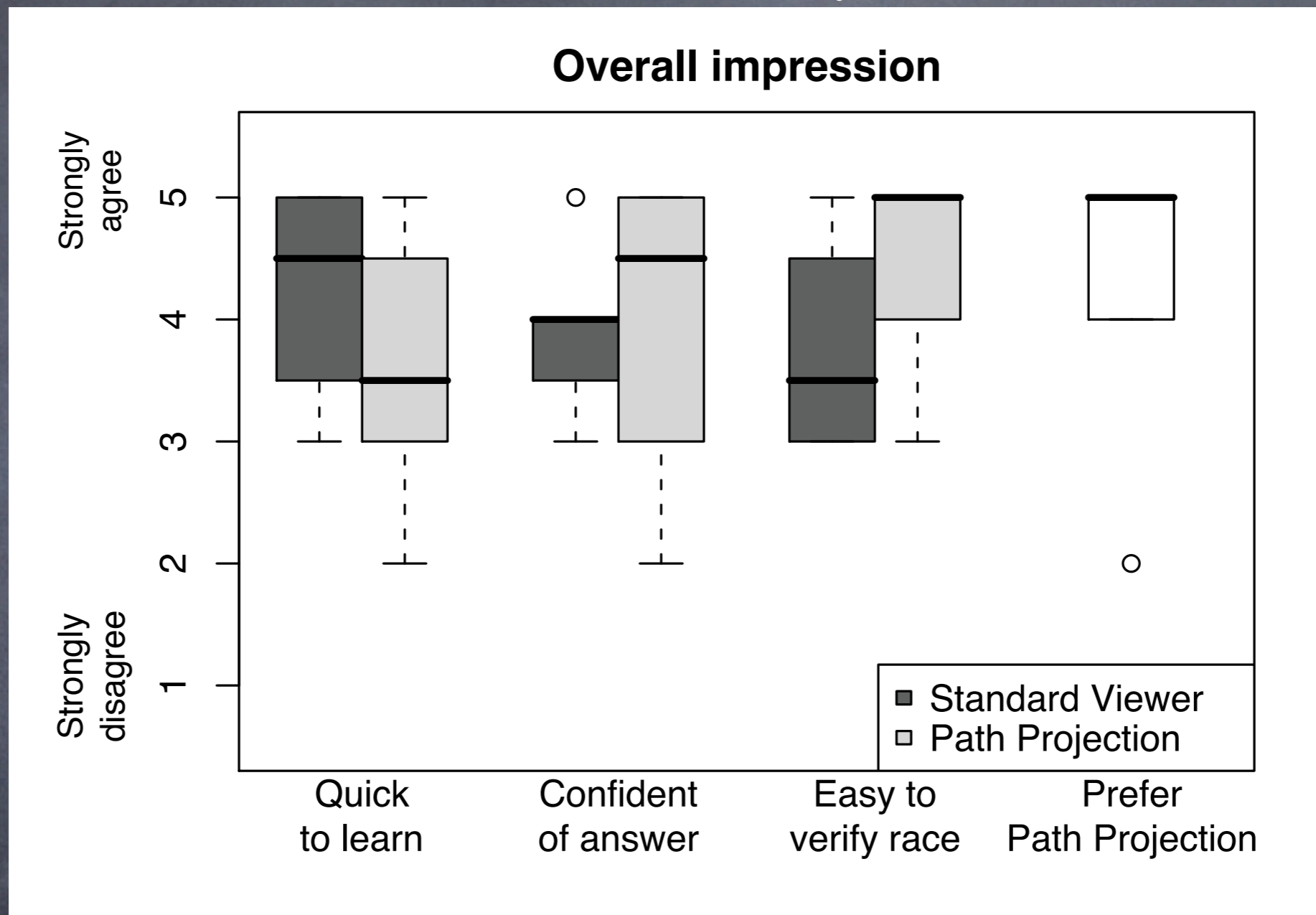
Less Use of Error Report

- Duration where pointer is over error report
(e.g., using hyperlinks)
- On average, only 20s with PP vs. 94s with SV*
- "Necessary for [SV], but just a convenience in [PP]."



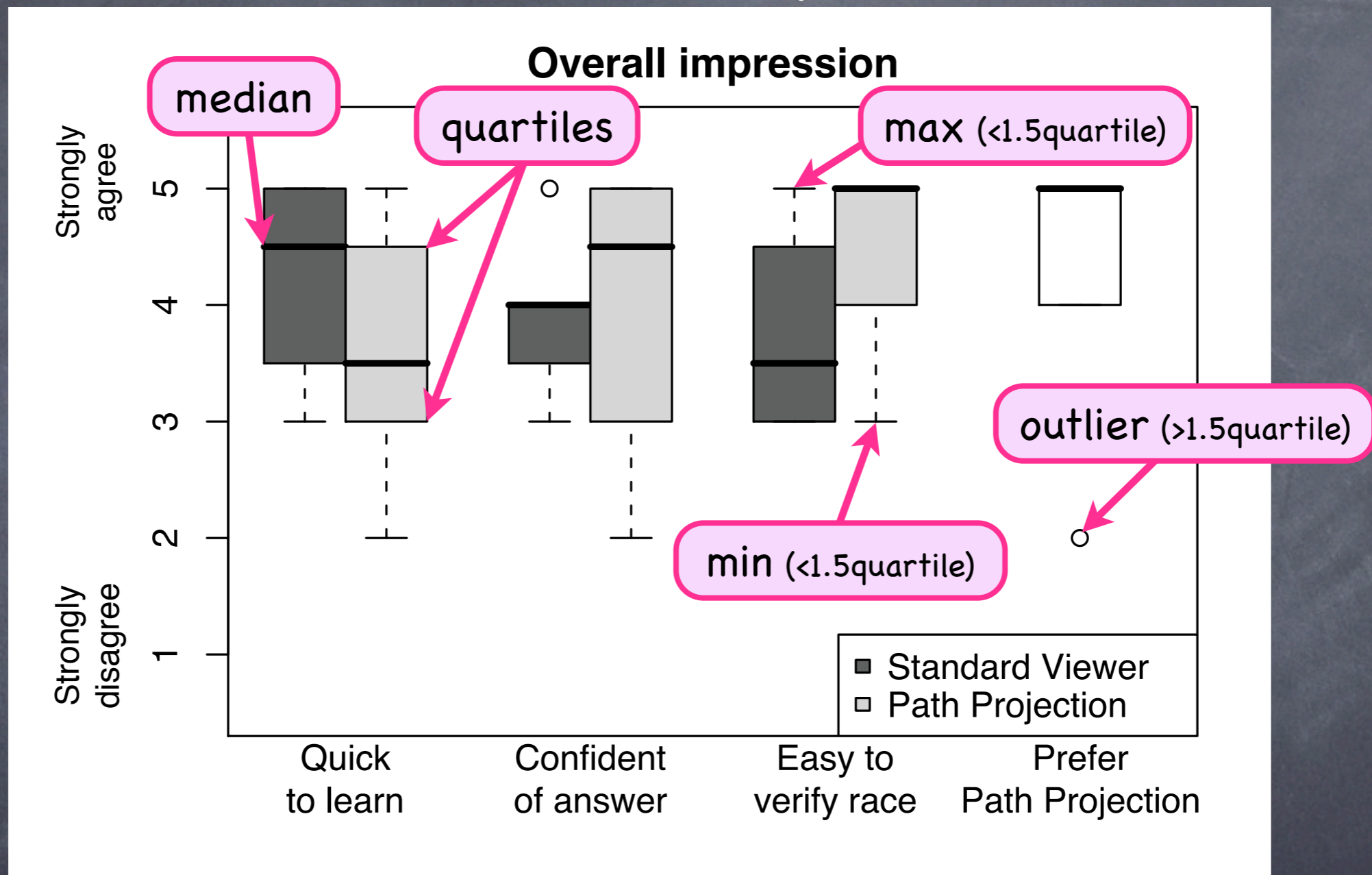
*statistically significant ($p < 0.05$)

Qualitative (Boxplot Guide)



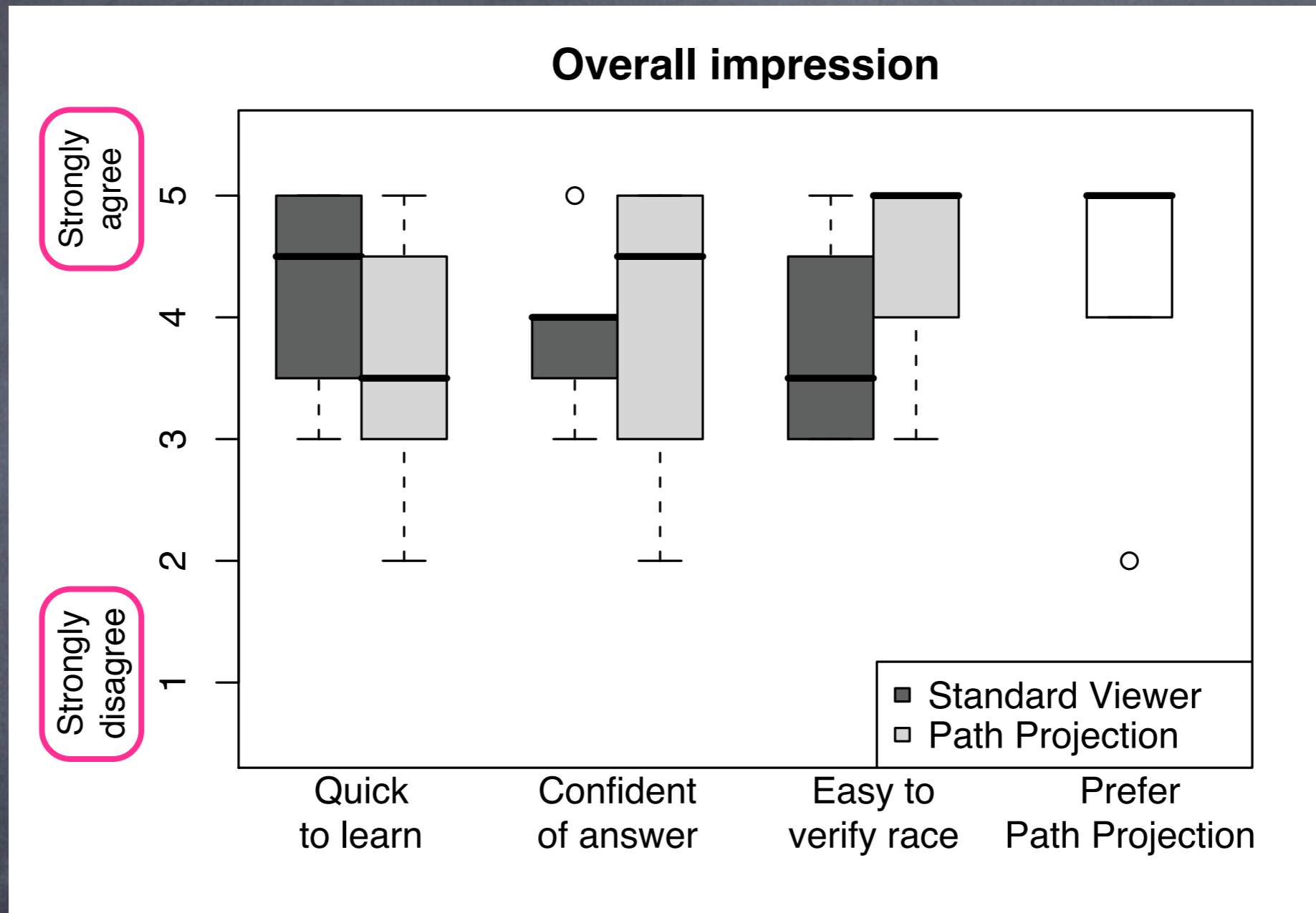
- We asked participants to rate on 1-5 scale

Qualitative (Boxplot Guide)

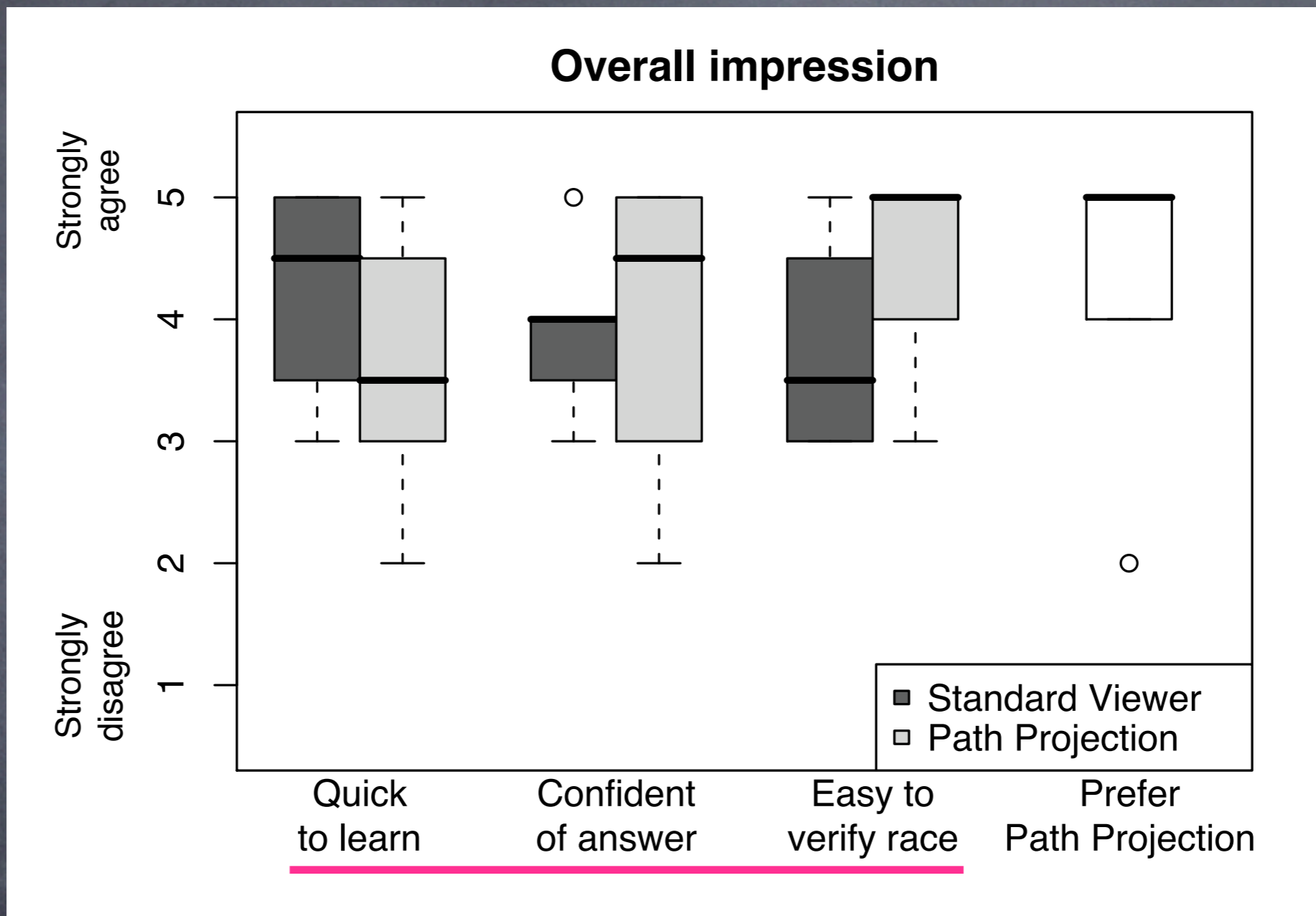


- We asked participants to rate on 1-5 scale
- Results summarized in boxplots

Prefer Path Projection

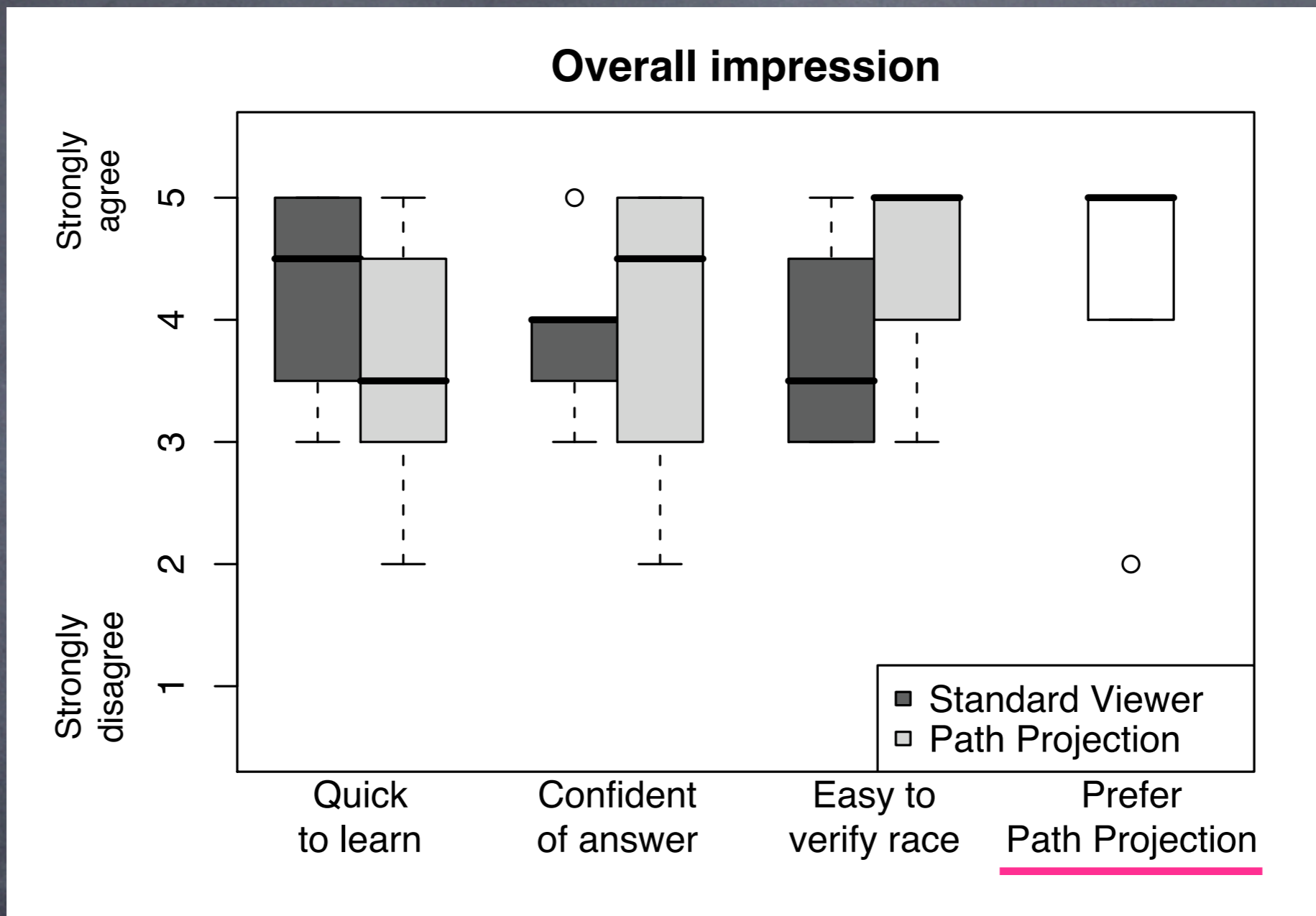


Prefer Path Projection



Quick/confident/easy: not statistically significant

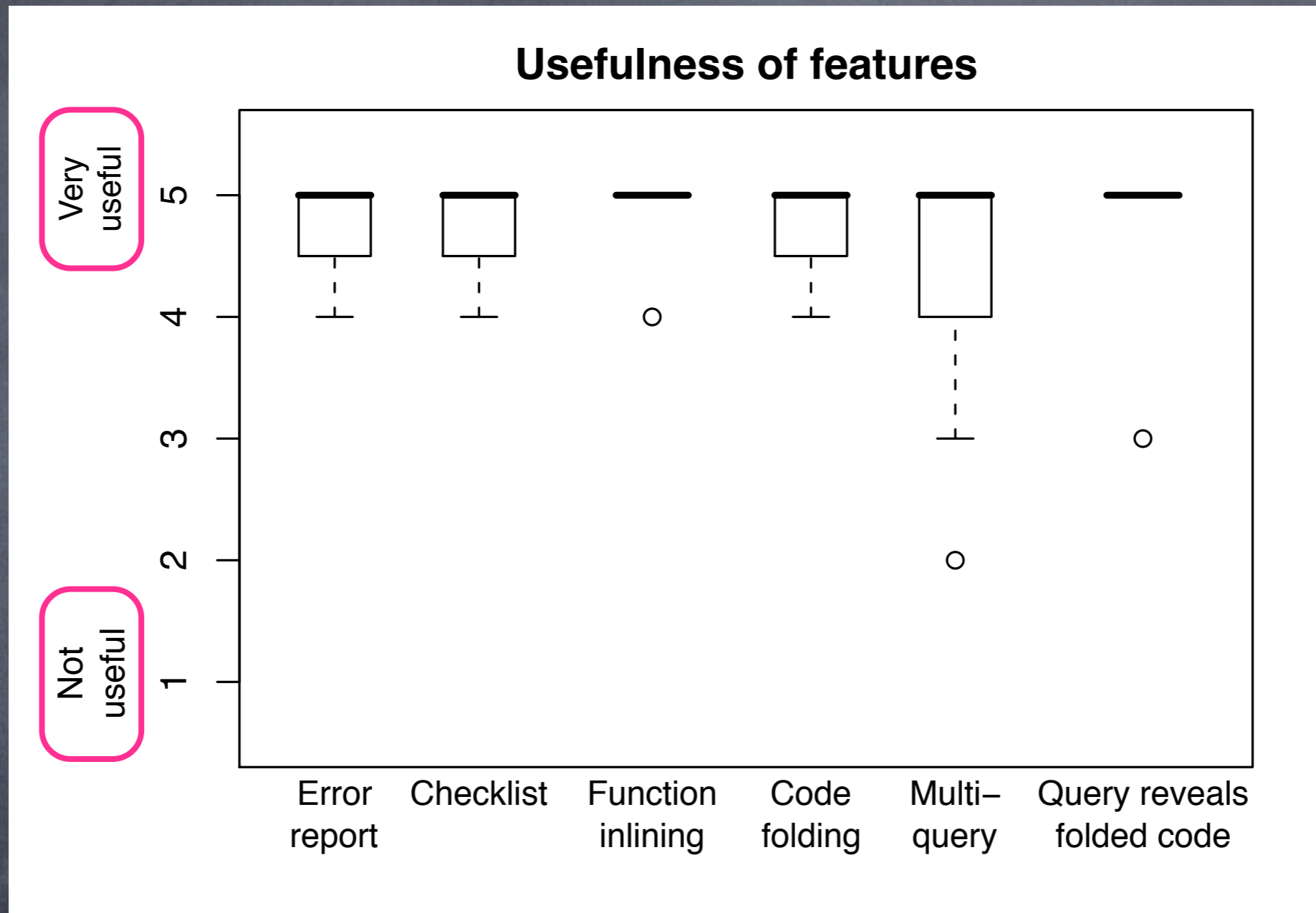
Prefer Path Projection



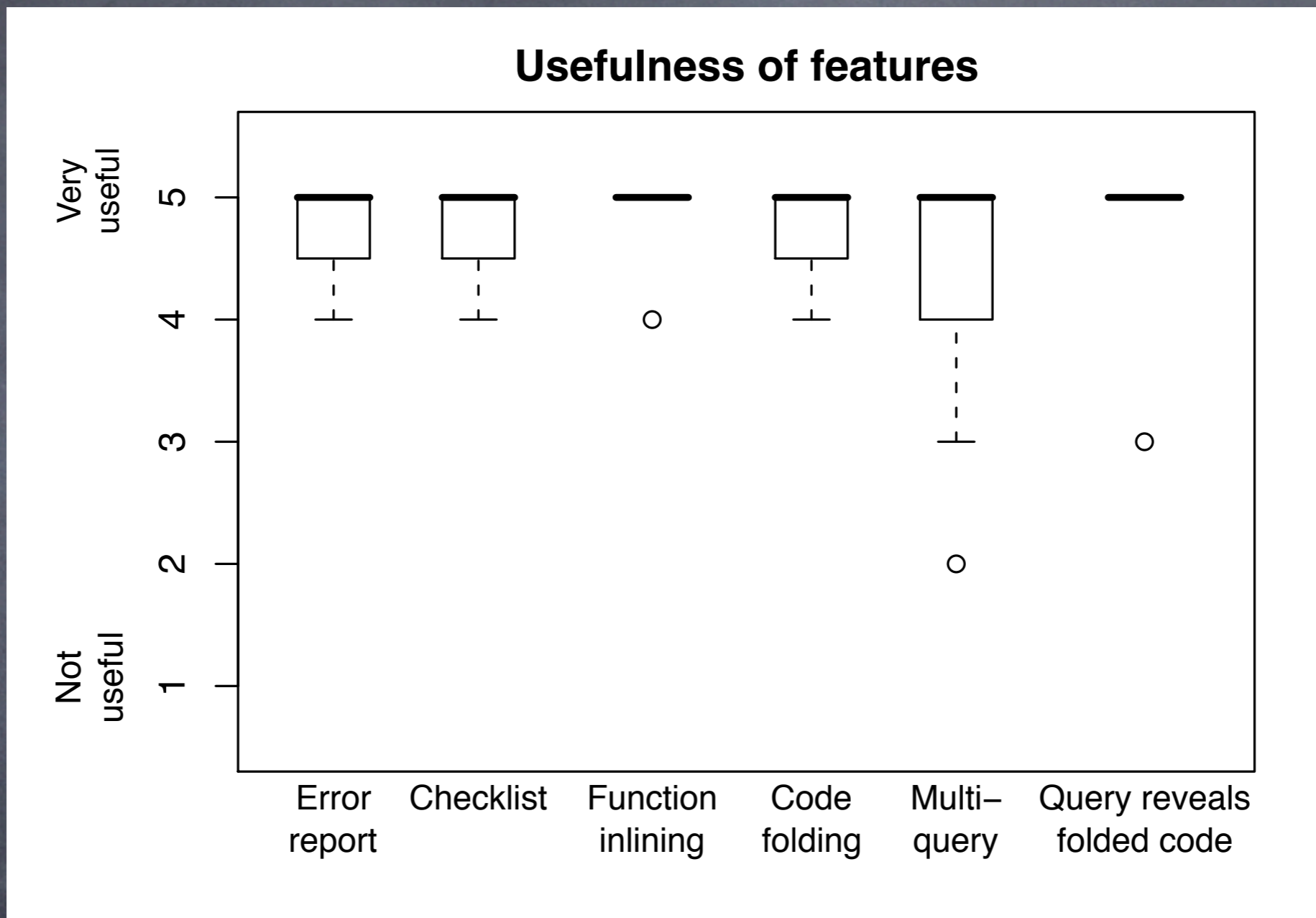
- Quick/confident/easy: not statistically significant
- Preference: all but one preferred PP to SV*

*statistically significant ($p < 0.05$)

Path Projection Features



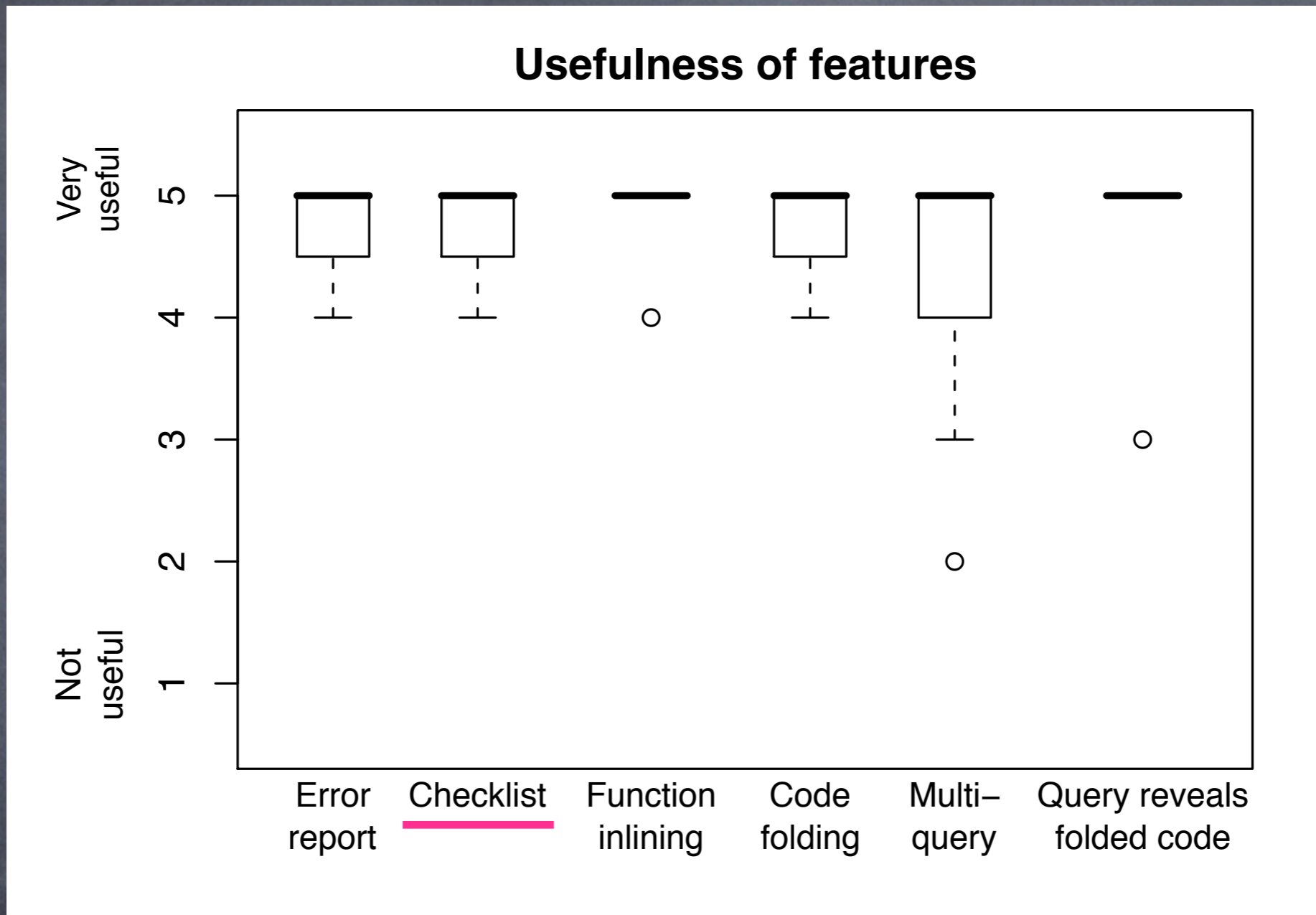
Path Projection Features



● Generally favorable towards PP features*

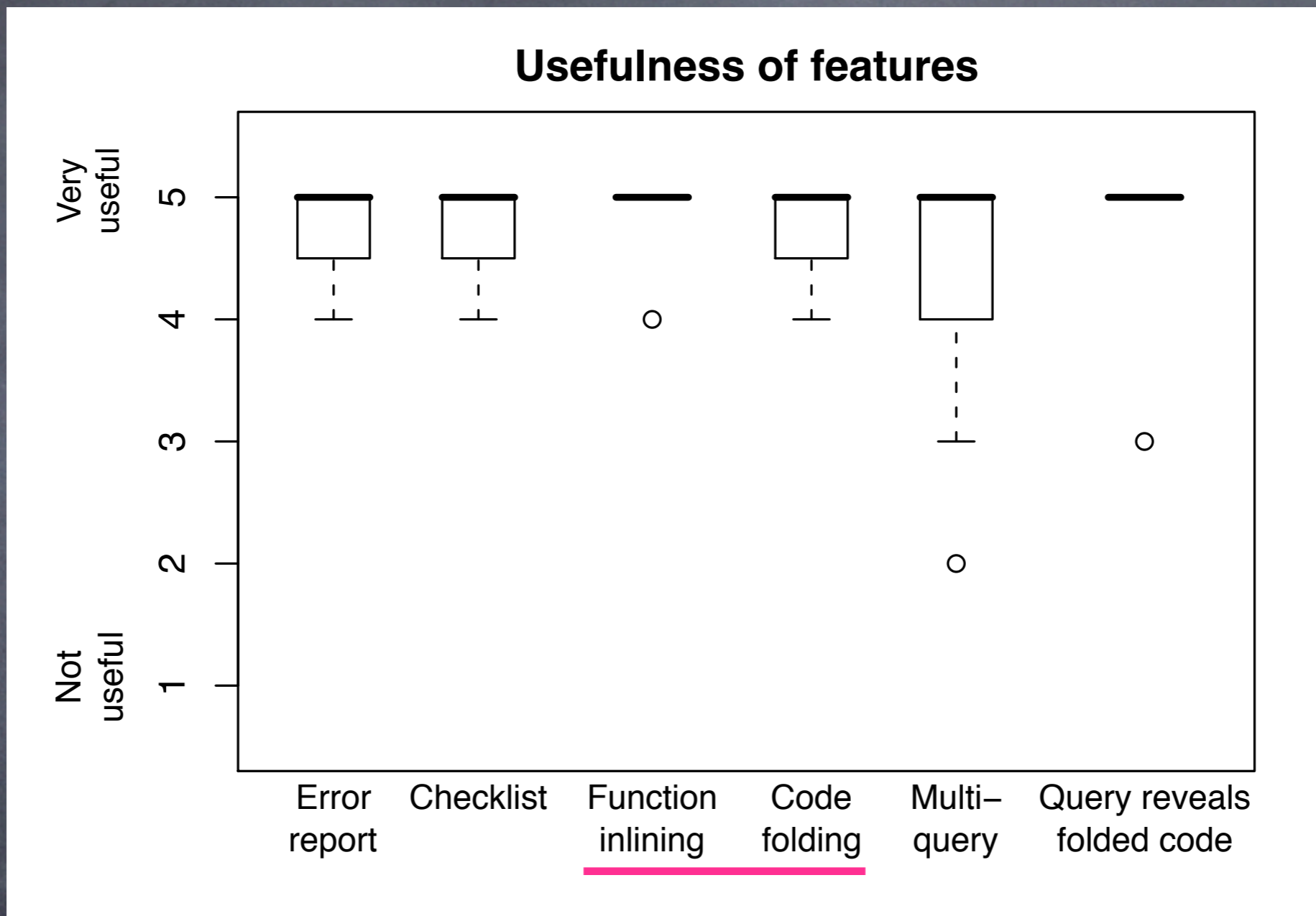
*statistically significant ($p < 0.05$)

Path Projection Features



- Checklist: "saved me from having to memorize rules"

Path Projection Features



- Checklist: "saved me from having to memorize rules"
- Surprisingly, favored function inlining/code folding code folding was "the best feature" or "my favorite feature"

Threats To Validity

- Experimental design limitations
 - small number of users and trials
 - not static analysis experts, unfamiliar programs
 - statistically significant despite limitations
- Standard Viewer not “real” editor
 - deliberate choice to avoid bias from prior experience
- The checklist might bias users
 - checklist designed for Locksmith, not SV or PP
 - both interfaces use the same checklist

Conclusion

- Path Projection: a new UI toolkit for visualizing program paths
- Can be used with any static analysis tools
 - Takes an XML path report as input
- Our study showed that it improves completion time (18%) with similar accuracy and users liked it
- Try it at: <http://www.cs.umd.edu/projects/PL/PP/>