

Bayou: Learning to Write Programs Using Big Code

Chris Jermaine
Department of Computer Science
Rice University

Classical problem in Artificial Intelligence

“Planning”

- Given a starting state, a set of possible actions to take, a description of the acceptable end states...
- Generate a set of actions that will get you to an acceptable end state...

Most General Form of the Planning Problem

“Program Synthesis”

- Given a high-level description for a desired computer program (inputs and outputs)
- An AI should automatically generate a program in a language such as Java, Python, C++ that matches the description

Compelling Problem — If We Can Solve It

- Programmers are expensive
- They are very hard to find (especially in Houston)
- And precious few of them are really good :-)

People Have Tried to Solve Synthesis Forever



- Problem first posed in 1957 by Alonzo Church
- Church: Alan Turing's advisor



People Have Tried to Solve Synthesis Forever



- Problem first posed in 1957 by Alonzo Church
- Church: Alan Turing's advisor

Not really Turing; just played him in "The Imitation Game" :-)



Why Is Program Synthesis So Hard?

- A Java/Python/C++ computer program can solve **every** problem that can be solved
 - At least, most theoreticians think so
 - Limited only by compute power of the computer it runs on
- So solving program synthesis is as hard as solving **every** problem **ever!**

In 60 Years, Not Much Serious Progress

- But now, we've suddenly got some hope
 - Microsoft, Google, Facebook all looking seriously at the problem
 - US DoD: MUSE project (\$13 million to Rice via Pliny)
- Why? What has changed?

Two Big Developments

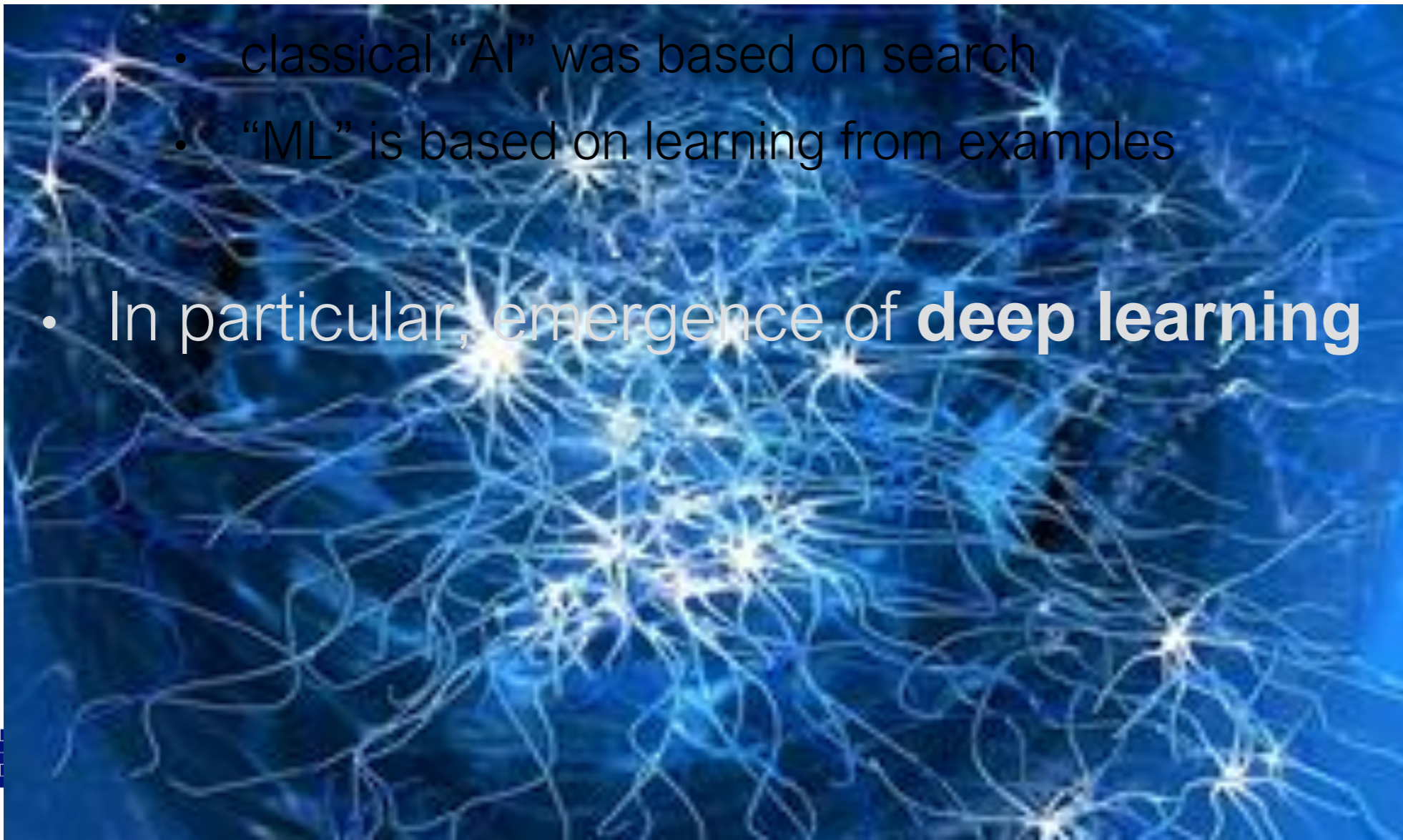
- First is shift from “Artificial Intelligence” to “Machine Learning”
 - classical “AI” was based on search
 - “ML” is based on learning from examples

Two Big Developments

- First is shift from “Artificial Intelligence” to “Machine Learning”

- classical “AI” was based on search
- “ML” is based on learning from examples

- In particular, emergence of **deep learning**



Two Big Developments

- Second is massive increase in open source software
 - 2007: perhaps 200K open source projects
 - 2015: perhaps 2M open source projects
- “Open source” ... code is freely available
- Means we have tons of data

Bayou-“Bayesian program synthesis for you”

- Ingest millions/billions of lines of code
 - Use a deep neural network to learn what programs typically look like
 - Expensive! Done offline
- Deep NN encodes a probability distribution over ***program sketches***

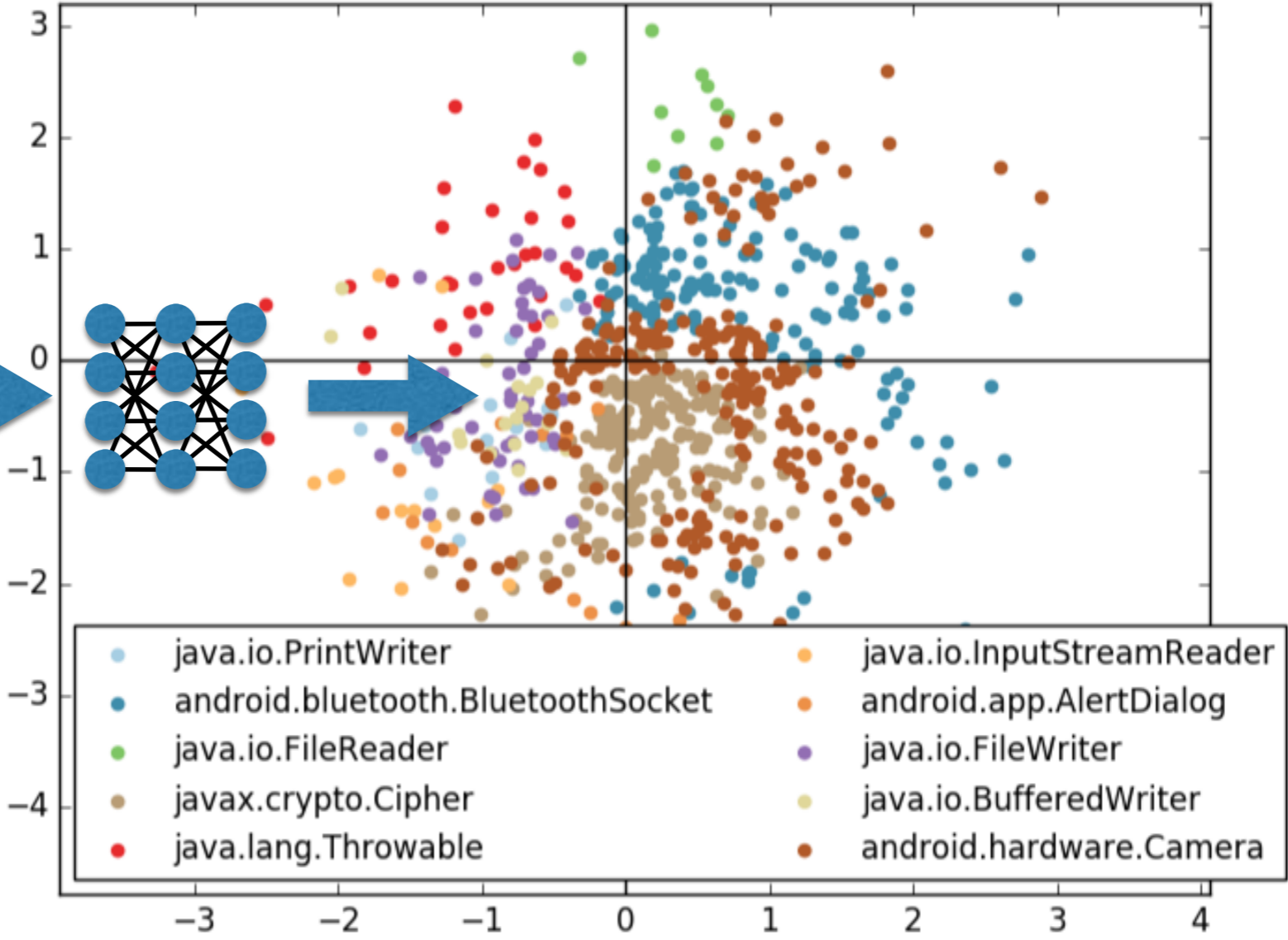
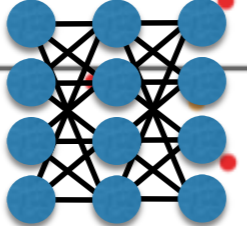
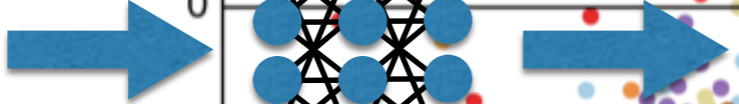
Bayou-“Bayesian program synthesis for you”

- To use this deep network, a human being gives Bayou clues or **evidence** as to the program s/he wants synthesized
 - Types to use... ex: “FileReader”
 - Functions to call... ex: “ReadLine”
 - Natural language: “Bayou, I want a program that is going to read from a file and print the contents to the screen”

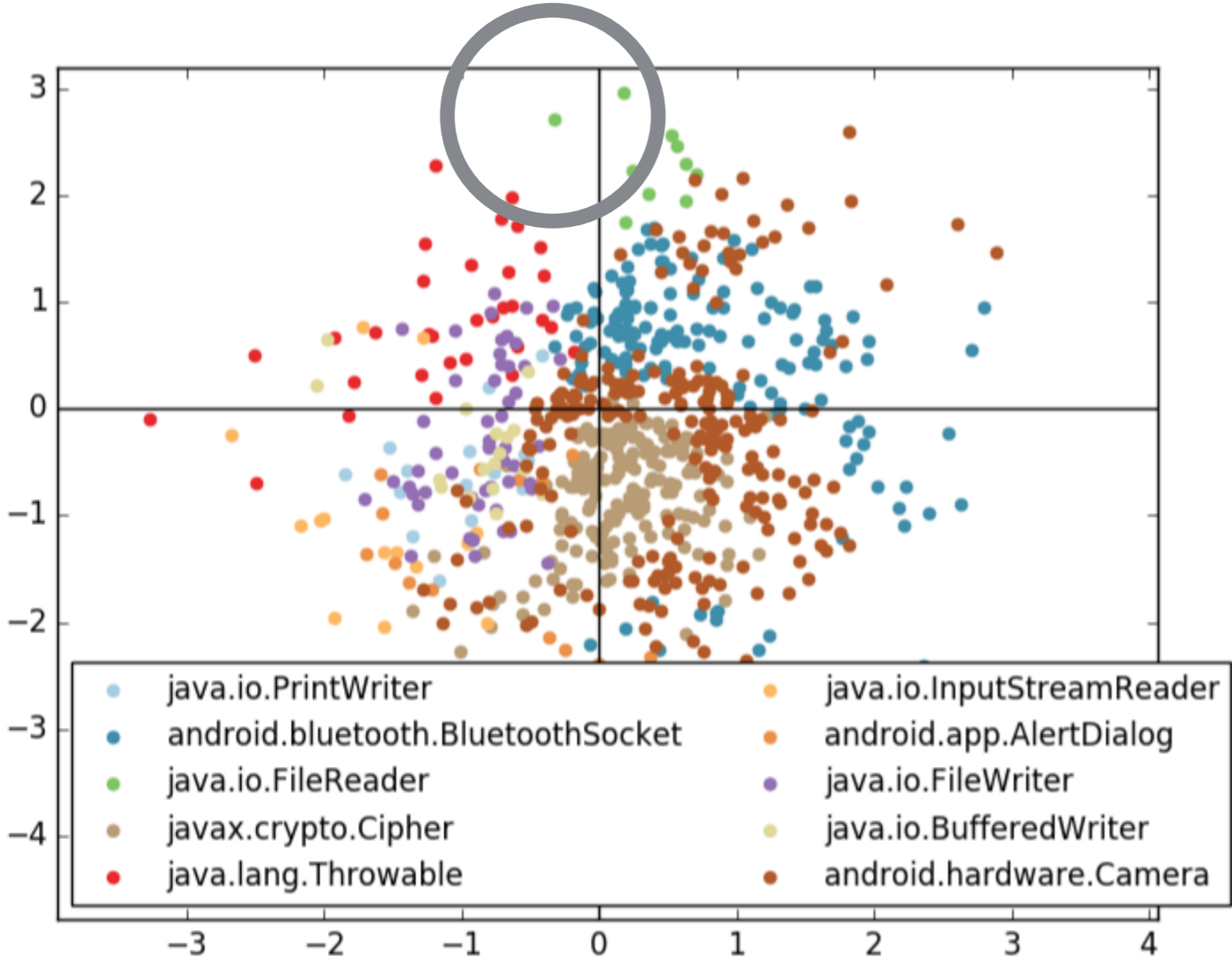
First a Deep Net Computes Posterior in Latent Space

```
public class TestIO {  
  void read(String file) {  
    ??  
  }  
}
```

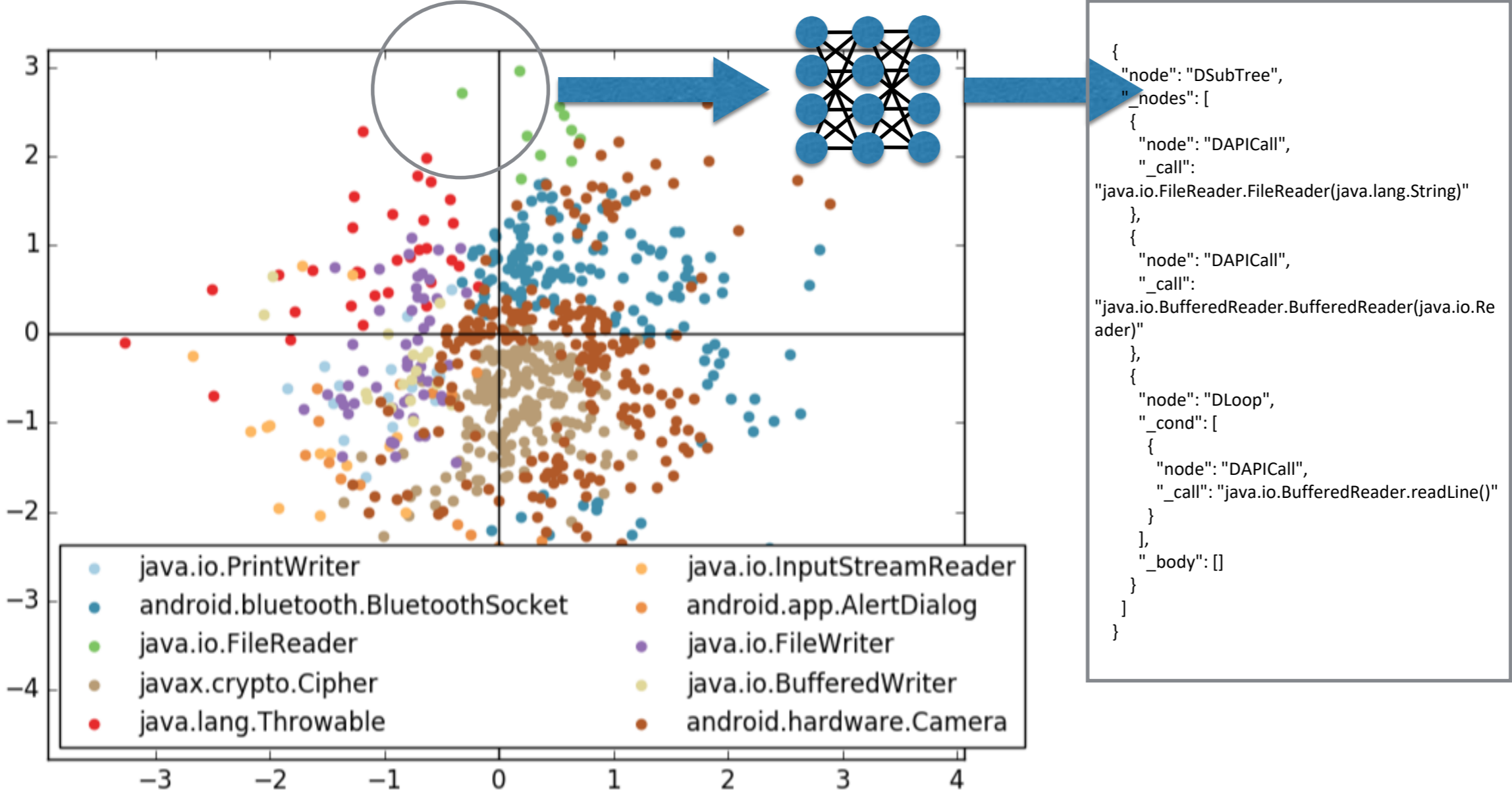
“Bayou, I want a program...”



Sample From Posterior



Deep Net Decodes Into a Sketch



Classical AI To Convert Sketch to Code

```
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.IOException;
public class TestIO {
    void read(String file) {
        String s;
        BufferedReader br;
        FileReader fr;
        try {
            fr = new FileReader(file);
            br = new BufferedReader(fr);
            while ((s = br.readLine()) != null) {
            }
        } catch (FileNotFoundException _e) {
        } catch (IOException _e) {
        }
    }
}
```

Symbolic
AI Search



```
{
  "node": "DSubTree",
  "_nodes": [
    {
      "node": "DAPICall",
      "_call":
      "java.io.FileReader.FileReader(java.lang.String)"
    },
    {
      "node": "DAPICall",
      "_call":
      "java.io.BufferedReader.BufferedReader(java.io.Re
ader)"
    },
    {
      "node": "DLoop",
      "_cond": [
        {
          "node": "DAPICall",
          "_call": "java.io.BufferedReader.readLine()"
        }
      ],
      "_body": []
    }
  ]
}
```

Will This End Programming?

- Eventually!
 - But it won't end software engineering

Will This End Programming?

- Eventually!
 - But it won't end software engineering
- But that's many decades away
 - Short term goal: tools that decrease programmer burden
 - Increase productivity

<http://www.askbayou.com>

Questions?