

Using an LLM to Help with Code Understanding



Daye Nam
dayen@cs.cmu.edu



Andrew Macvean
amacvean@google.com



Vincent Hellendoorn
vhellendoorn@cmu.edu



Bogdan Vasilescu
vasilescu@cmu.edu



Brad Myers
bam@cs.cmu.edu



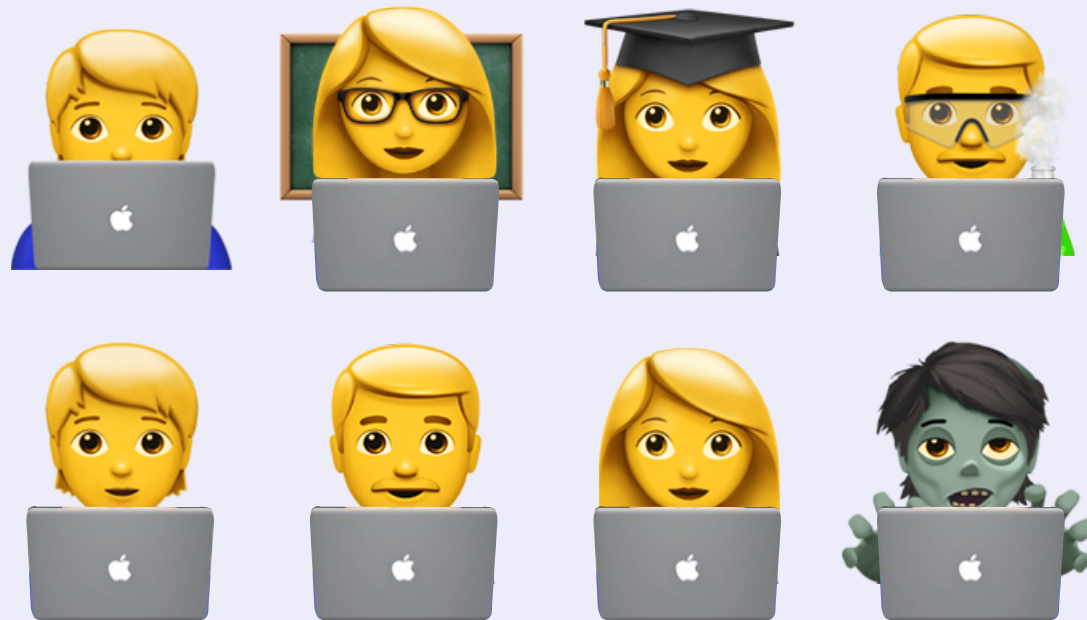
Carnegie Mellon University
School of Computer Science



Human-
Computer
Interaction
Institute



Programmers often need to work with
unfamiliar code written by other programmers



Programmers now also need to work with *unfamiliar* code written by AIs



Working with unfamiliar code is challenging



Hard-to-Answer Questions about Code (LaToza and Myers, PLATEAU' 10)
Questions Programmers Ask During Software Evolution Tasks (Sillito et al., FSE'14)

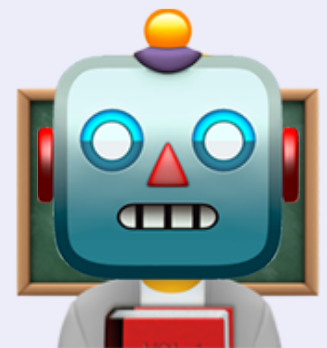
- ? What is the purpose of this code?
- ? What code could have caused this behavior?
- ? Which type represents this domain concept?
- ? ...



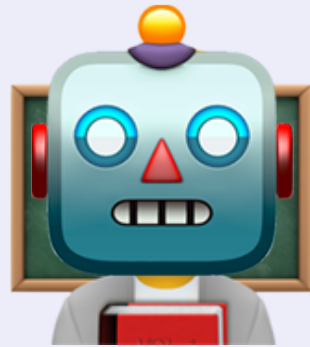
Can AIs also help programmers understand the unfamiliar code written by others?



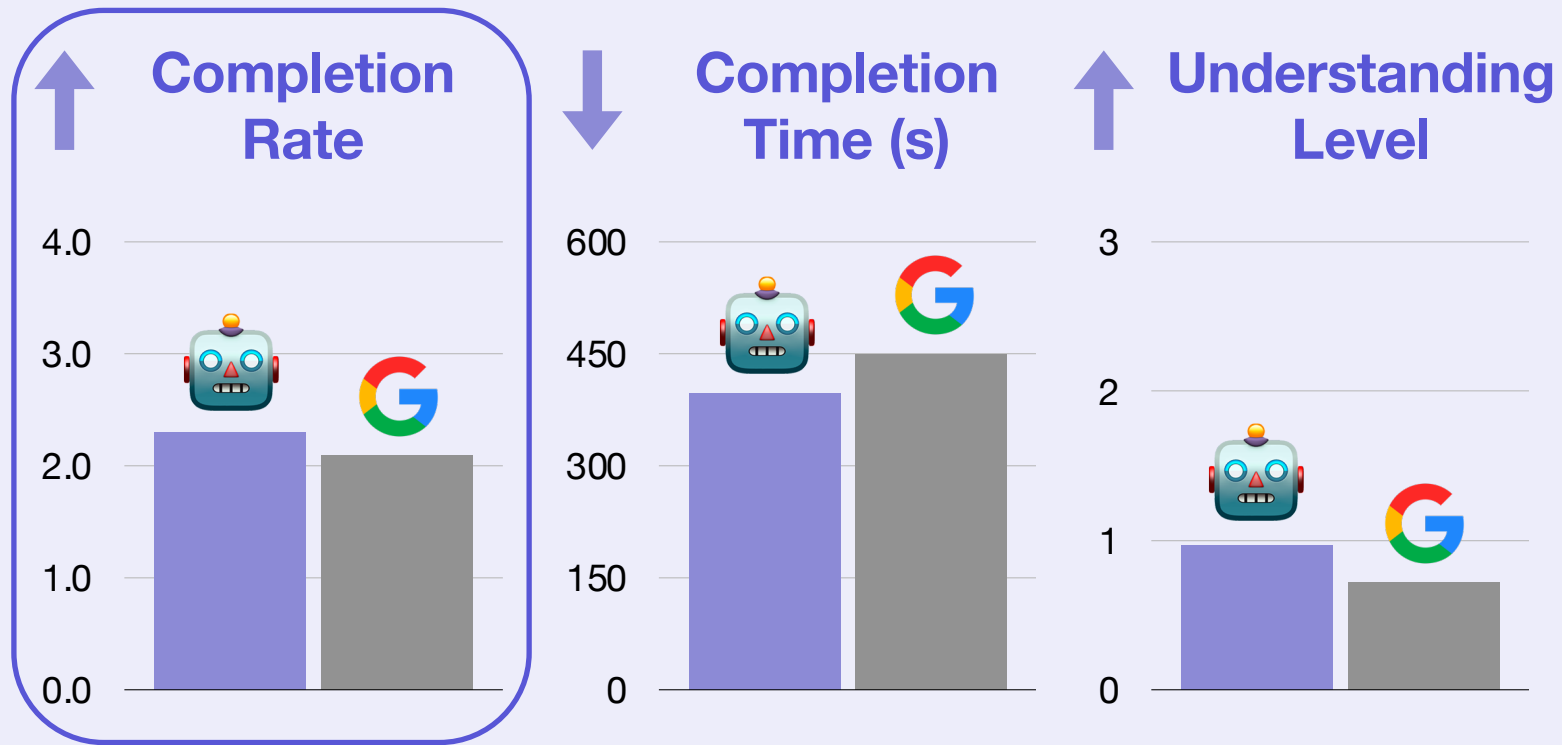
💡 This code loads a 3D mesh of a bunny, computes vertex normals ...



Hypothesis: Carefully designed AI assistant helps programmers work with unfamiliar code



Spoiler: AI can help users working with unfamiliar code



Statistically significant!

Hypothesis: Carefully designed AI assistant helps programmers work with unfamiliar code

The screenshot shows the GILT AI assistant interface. On the left is a sidebar with icons for various functions. The main area is a code editor displaying a Python script named 'open3d_task.py'. The script imports 'open3d', 'numpy', and 'matplotlib.pyplot', and contains logic for reading a point cloud, clustering it with DBSCAN, and visualizing the results. At the bottom, there is a chat window with the prompt 'Where do I move bu' and an 'Ask' button. The status bar at the bottom indicates 'Ln 5, Col 16' and 'Python 3.9.6 64-bit'.



LLM-based plugin for code understanding : In-IDE

```
1 import open3d as o3d
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     pcd_point_cloud = o3d.data.DemoCropPointCl
7     pcd = o3d.io.read_point_cloud(pcd_point_cl
8
9
10     with o3d.utility.VerbosityContextManager(
11         o3d.utility.VerbosityLevel.Debug) as c
12         labels = np.array(
13             pcd.cluster_dbscan(eps=0.1, min_po
14         )
15
16     max_label = labels.max()
17     print(f"point cloud has {max_label + 1} cl
18     colors = plt.get_cmap("Accent")(labels / (
19     colors[labels < 0] = 0
20     pcd.colors = o3d.utility.Vector3dVector(co
21
22     bunny = o3d.data.BunnyMesh()
23     bunny_mesh = o3d.io.read_triangle_mesh(bun
24     bunny_mesh.compute_vertex_normals()
25     bunny_mesh.transform([[[-1, 0, 0, 0],
26                             [0, 1, 0, 0],
27                             [0, 0, 1, 0],
28                             [0, 0, 0, 1]])
29     bunny_mesh.translate([2.5, 2.1, 1.2])
```

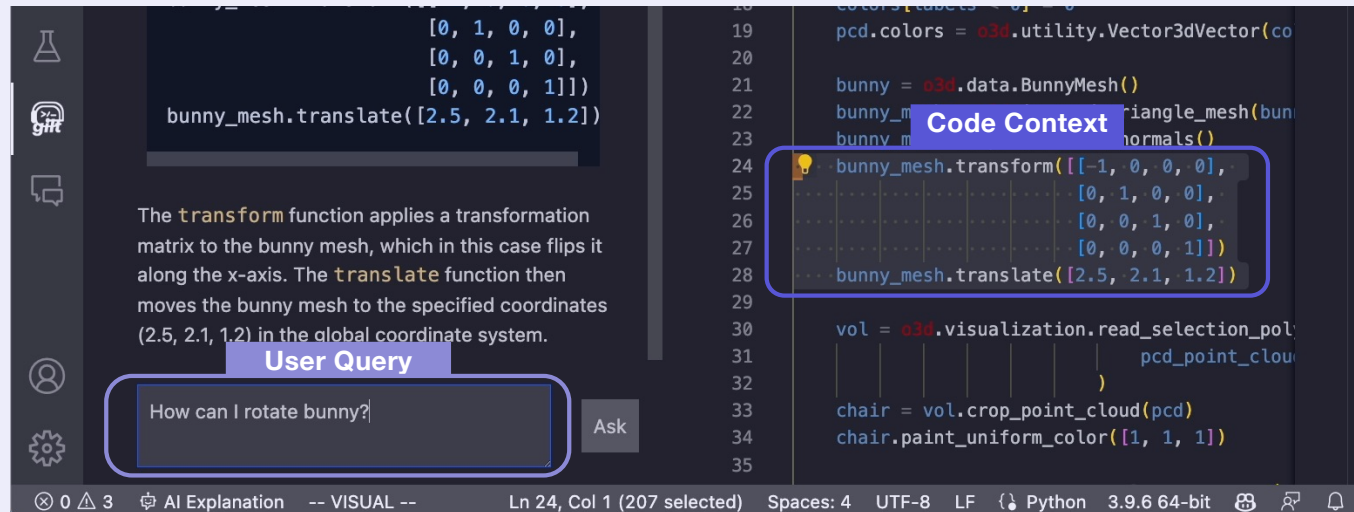
Where do I move bu

Ask

0 3 AI Explanation -- NORMAL -- Ln 5, Col 16 Spaces: 4 UTF-8 LF Python 3.9.6 64-bit



LLM-based plugin for code understanding : Context-aware

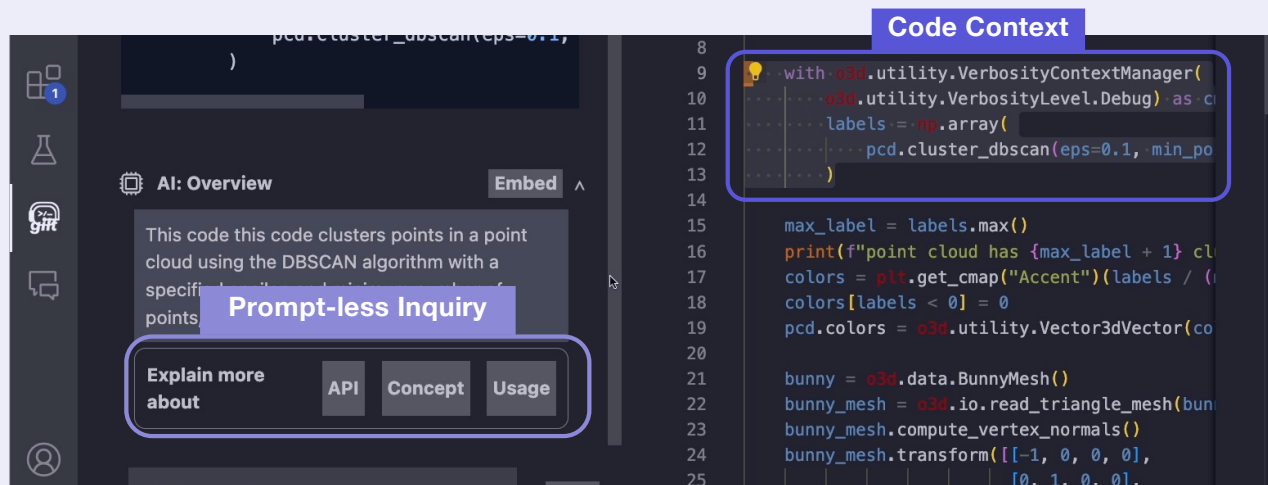


In the context of the following code,

{ **Code Context** }, { **User Query** }?



LLM-based plugin for code understanding : Prompt-less



A Theory of Robust API Knowledge (Thayer et al. TCE'2021)

*Please provide an **Open3d** code example
mainly showing the usage of the following API calls:*

*Extracted from the
code context*

BunnyMesh, read triangle mesh, compute vertex normals, transform, translate:

Code Context



LLM-based plugin for code understanding

```
1 import open3d as o3d
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     pcd_point_cloud = o3d.data.DemoCropPointCloud()
7     pcd = o3d.io.read_point_cloud(pcd_point_cloud.path)
8
9     with o3d.utility.VerbosityContextManager(
10         o3d.utility.VerbosityLevel.Debug) as cm:
11         labels = np.array(
12             pcd.cluster_dbscan(eps=0.1, min_points=10)
13         )
14
15     max_label = labels.max()
16     print(f"point cloud has {max_label + 1} clusters")
17     colors = plt.get_cmap("Accent")(labels / (max_label + 1))
18     colors[labels < 0] = 0
19     pcd.colors = o3d.utility.Vector3dVector(colors)
20
21     bunny = o3d.data.BunnyMesh()
22     bunny_mesh = o3d.io.read_triangle_mesh(bunny.path)
23     bunny_mesh.compute_vertex_normals()
24     bunny_mesh.transform([[-1, 0, 0, 0],
25                          [0, 1, 0, 0],
26                          [0, 0, 1, 0],
27                          [0, 0, 0, 1]])
28     bunny_mesh.translate([2.5, 2.1, 1.2])
```

Where do I move bu

Ask

```
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     pcd_point_cloud = o3d.data.DemoCropPointCl
7     pcd = o3d.io.read_point_cloud(pcd_point_cl
8
9     with o3d.utility.VerbosityContextManager(
10         o3d.utility.VerbosityLevel.Debug) as c
11         labels = np.array(
12             pcd.cluster_dbscan(eps=0.1, min_po
13         )
14
15     max_label = labels.max()
16     print(f"point cloud has {max_label + 1} cl
17     colors = plt.get_cmap("Accent")(labels / (
18     colors[labels < 0] = 0
19     pcd.colors = o3d.utility.Vector3dVector(co
20
21     bunny = o3d.data.BunnyMesh()
22     bunny_mesh = o3d.io.read_triangle_mesh(bun
23     bunny_mesh.compute_vertex_normals()
24     bunny_mesh.transform([[-1, 0, 0, 0],
25                          [0, 1, 0, 0],
26                          [0, 0, 1, 0],
27                          [0, 0, 0, 1]])
28     bunny_mesh.translate([2.5, 2.1, 1.2])
```



LLM-based plugin for code understanding

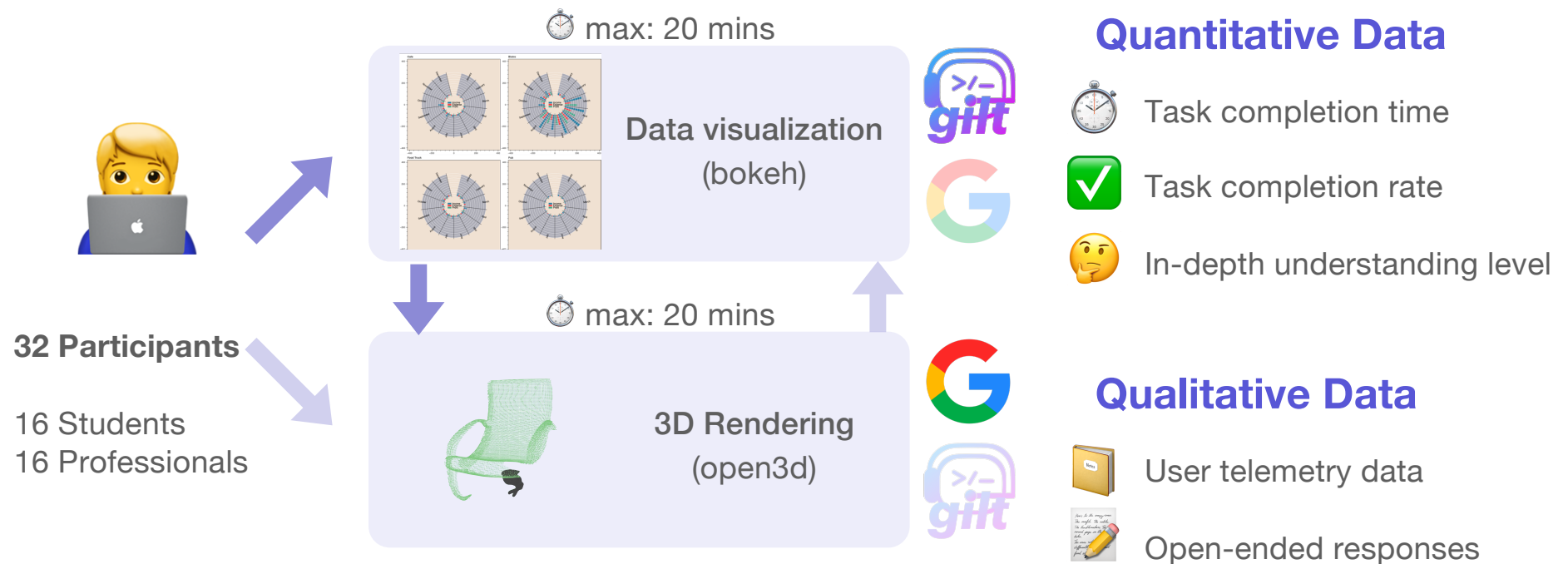
```
1 import open3d as o3d
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     pcd_point_cloud = o3d.data.DemoCropPointCloud()
7     pcd = o3d.io.read_point_cloud(pcd_point_cloud.path)
8
9     with o3d.utility.VerboseContextManager(
10         o3d.utility.VerboseLevel.Debug) as cm:
11         labels = np.array(
12             pcd.cluster_dbscan(eps=0.1, min_points=10)
13         )
14
15     max_label = labels.max()
16     print(f"point cloud has {max_label + 1} clusters")
17     colors = plt.get_cmap("Accent")(labels / (max_label + 1))
18     colors[labels < 0] = 0
19     pcd.colors = o3d.utility.Vector3dVector(colors)
20
21     bunny = o3d.data.BunnyMesh()
22     bunny_mesh = o3d.io.read_triangle_mesh(bunny.path)
23     bunny_mesh.compute_vertex_normals()
24     bunny_mesh.transform([[-1, 0, 0, 0],
25                          [0, 1, 0, 0],
26                          [0, 0, 1, 0],
27                          [0, 0, 0, 1]])
28     bunny_mesh.translate([2.5, -2.1, 1.2])
```

Where do I move bu

Ask

Ln 5, Col 16 Spaces: 4 UTF-8 LF Python 3.9.6 64-bit

Study design: Web Search vs. GILT



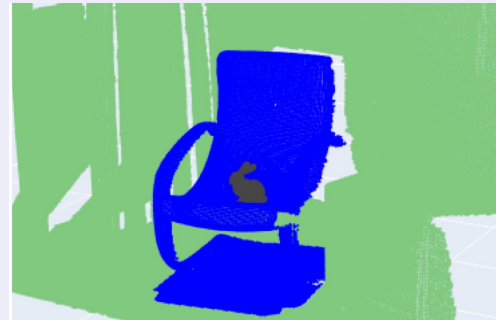
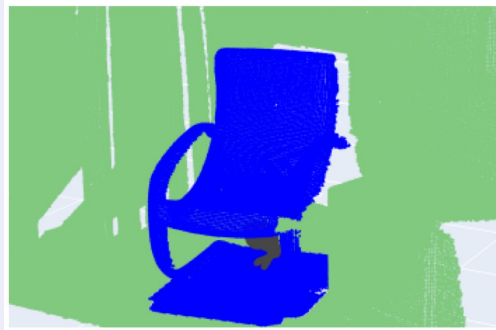
Example Sub-task: 3D Rendering (Open3D)



Task completion time



Task completion rate



Subtask 2: “Make the bunny sit upright on the chair”



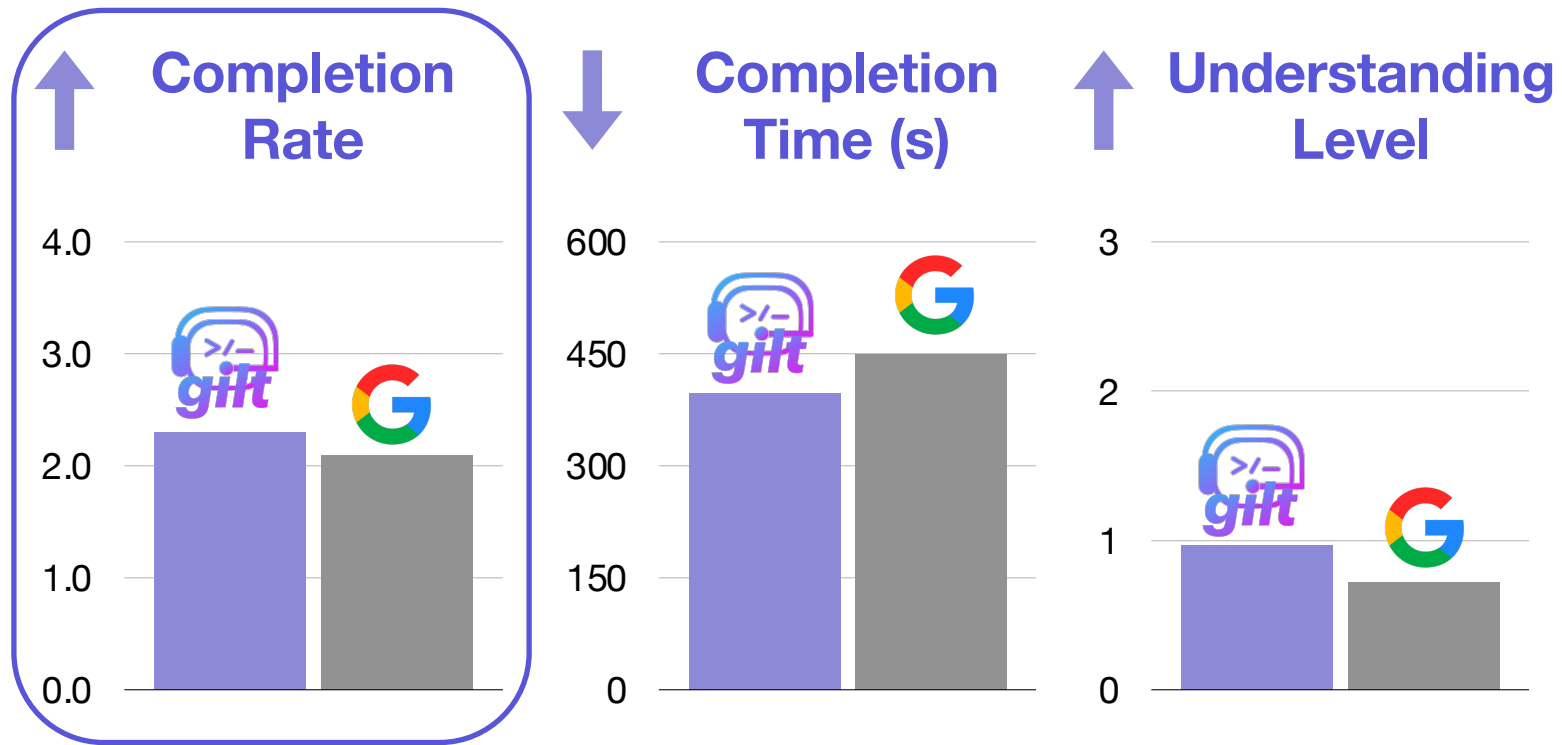
In-depth understanding level

```
bunny_mesh.transform([[-1, 0, 0, 0], [0, 1, 0, 0],  
                     [0, 0, 1, 0], [0, 0, 0, 1]])
```

Q2: What transformation does the 4x4 matrix do in the following code (i.e., what will happen to the bunny)?

GILT enhances users' ability to complete tasks

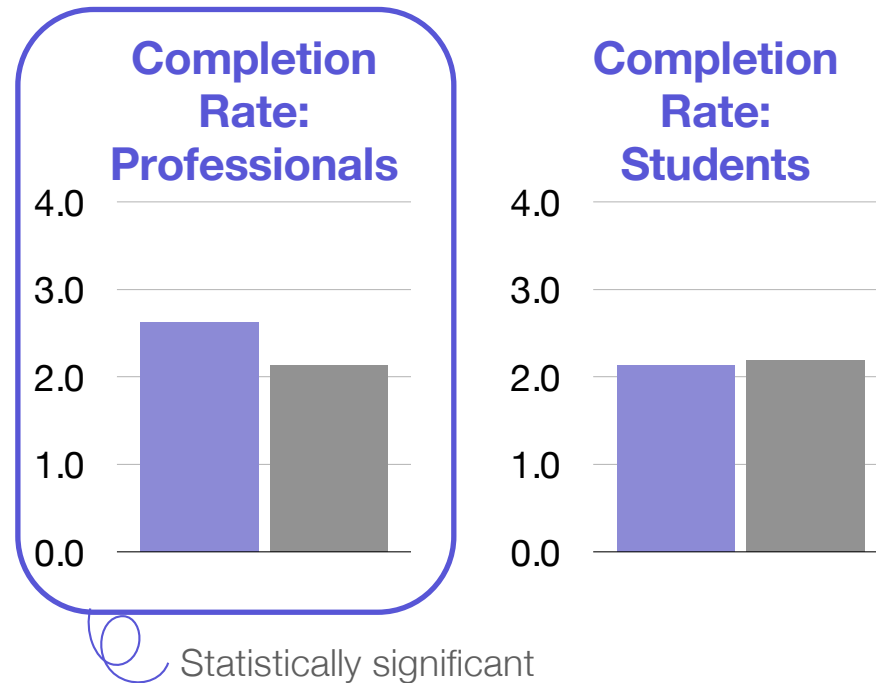
Control: domain experience, programming experience, familiarity with AI tools



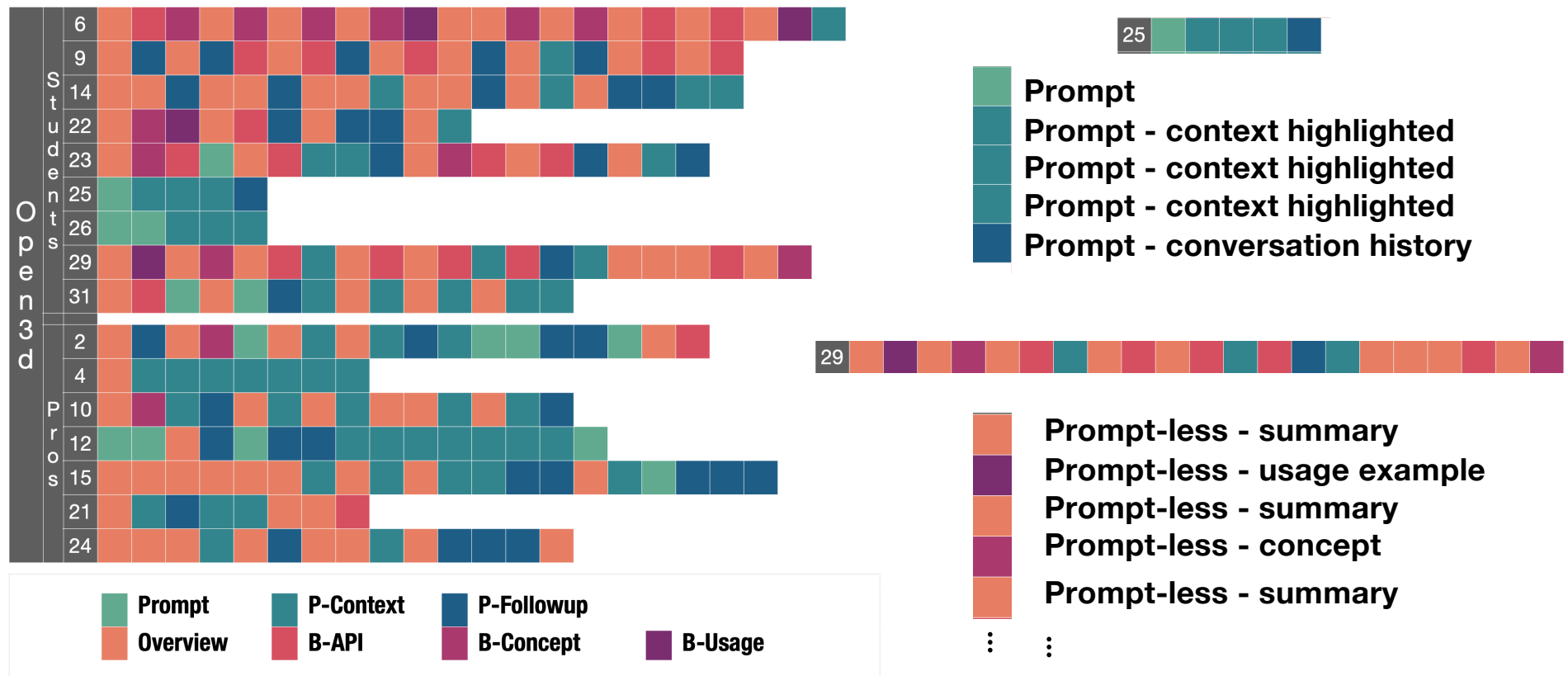
Statistically significant

The usage and benefits differ between students vs. professionals

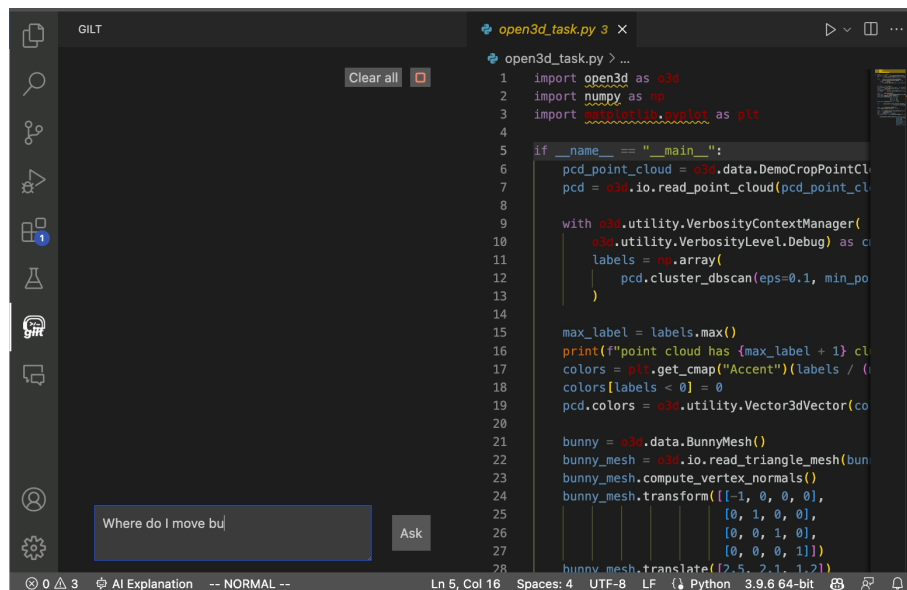
Control: domain experience, programming experience, familiarity with AI tools



The way participants interact with GILT varies



Using an LLM to Help with Code understanding



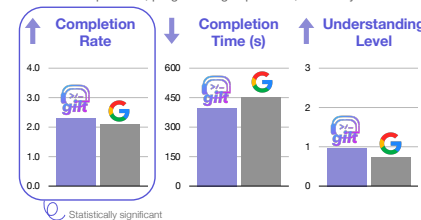
```
open3d_task.py 3 x
1 import open3d as o3d
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     pcd_point_cloud = o3d.data.DemoCropPointCloud()
7     pcd = o3d.io.read_point_cloud(pcd_point_cloud.path)
8
9     with o3d.utility.VerboseContextManager(
10         o3d.utility.VerboseLevel.Debug) as cm:
11         labels = np.array(
12             pcd.cluster_dbscan(eps=0.1, min_points=10))
13
14     max_label = labels.max()
15     print(f"point cloud has {max_label + 1} clusters")
16     colors = plt.get_cmap("Accent")(labels / (max_label + 1))
17     colors[labels < 0] = 0
18     pcd.colors = o3d.utility.Vector3dVector(colors)
19
20     bunny = o3d.data.BunnyMesh()
21     bunny_mesh = o3d.io.read_triangle_mesh(bunny.path)
22     bunny_mesh.compute_vertex_normals()
23     bunny_mesh.transform([[-1, 0, 0, 0],
24                           [0, 1, 0, 0],
25                           [0, 0, 1, 0],
26                           [0, 0, 0, 1]])
27     bunny_mesh.translate([2.5, 2.1, 1.2])
28     o3d.visualization.draw_geometries([pcd, bunny_mesh])
```

Where do I move bu

Ask

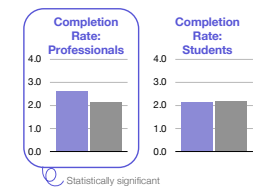
GILT enhances users' ability to complete tasks

Control: domain experience, programming experience, familiarity with AI tools

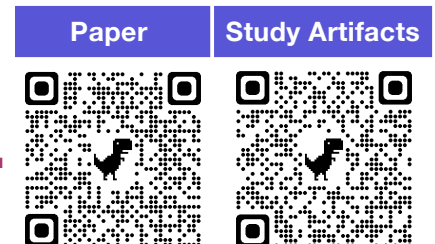
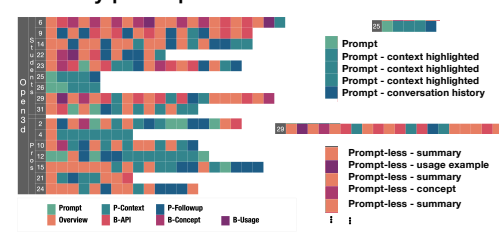


The usage and benefits differ between students vs. professionals

Control: domain experience, programming experience, familiarity with AI tools



The way participants interact with GILT varies



Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, Brad Myers

<https://dayenam.com>



Carnegie Mellon University
School of Computer Science

