

# CS 846 Software Engineering for Big Data and AI

## Grounded Copilot: How Programmers Interact with Code-Generating Models

Jun Lim

20870249

Recent advances in code-generating models have captured the attention of software developers. Tools like Github Copilot have proven to have impressive performance in assisting programmers with coding tasks. This paper presents a grounded theory analysis of how programmers interact with AI programming assistants like Github Copilot. The authors made observations of 20 participants of various levels of experience using the assistant to tackle diverse programming tasks and discovered the following modes of interaction:

### Acceleration mode

- In this mode, programmers have a clear roadmap and utilize the Copilot to reach their coding objectives faster.
- This assistant is seen as an intelligent auto-completion tool that helps complete a programmer's planned code actions with code snippets.
- Programmers also tend to accept Copilot's suggestions while maintaining their focus and thought process.
- The suggested code here is validated quickly through "pattern matching" of certain keywords and structures.
- There are also some drawbacks, such as longer suggestions that can disrupt the programmer's flow and focus.

### Exploration mode

- In this mode, programmers are unsure about how to proceed and use Copilot to explore their options.
- This assistant is seen as a "tutor", offering possible coding ideas, syntax guidance, suggesting APIs, or even entire task completion.
- Programmers rely on Copilot to help plan their code actions and learn about the unknown aspects of the tasks.
- Programmers in this mode are often willing to explore multiple options suggested by the copilot to help expand the programmer's understanding of their tasks.

Expanding from that study, the authors provided a few recommendations on improving the AI tool in the aspect of input, such as allowing better control over the prompt context, having cross-language translation, and in the aspect of output, such as having an indicator of the interaction mode that they're in, always-on code validation, etc.

In conclusion, the authors provided a comprehensive analysis of how AI is reshaping coding practices and advocated improvements to AI tools in the future.

### Paper Commentary

The paper provided a good analysis of the usage of Copilot, giving researchers an understanding of how the tool can be improved in the future. The authors have also utilized a credible approach - "grounded theory" to conduct the research, making the paper's findings to be trustworthy. Given the high popularity of AI, the paper provided a good foundation and starting point for future research in AI tools.

There are also some downsides to the paper. First, the authors used only 20 participants to conduct the study; this makes the study have limited generalizability as it doesn't capture the full demographics. Furthermore, the study was done with limited tasks (only writing code), and it doesn't highlight how the tool is used in cases like debugging or testing.