# Agent's Adventures in Coding Land

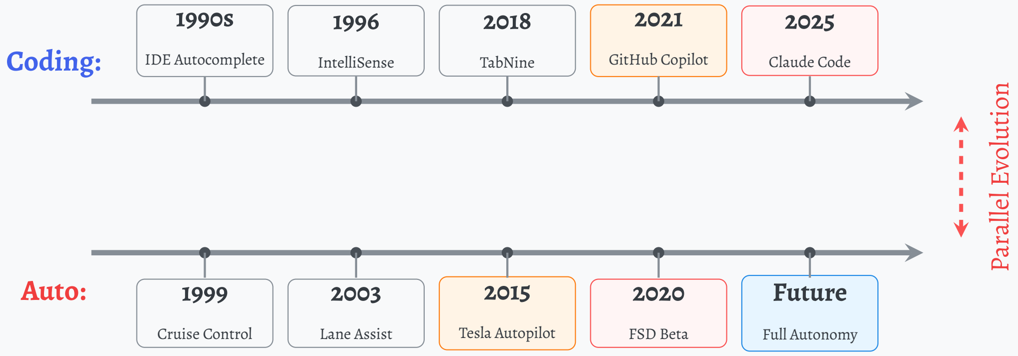
Intro to Claude Code and Codex

*September 13, 2025*

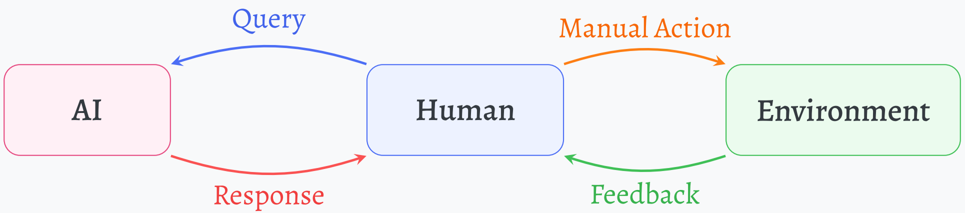
**Wenhao Chai**

Princeton University

# Evolution of AI Assistance



## Traditional Human–AI–Environment Interaction



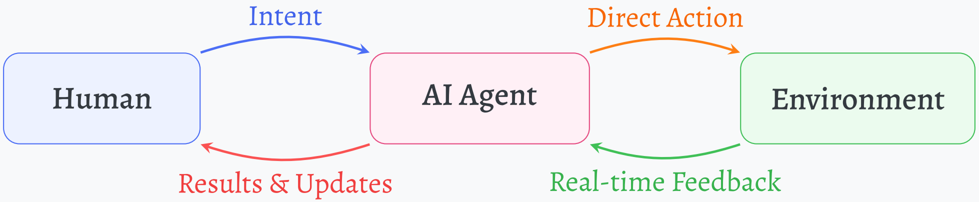
*Humans act as intermediaries: they receive AI responses, perform actions in the environment, and then relay feedback to the AI.*

## Autocomplete: The Beginning of AI-Assisted Coding

**PyTorch Neural Network Autocomplete**

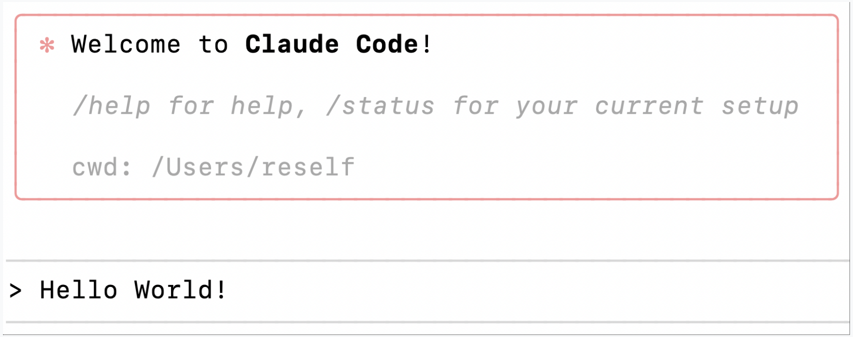
```
 # User types: "model = nn.Tra" + Ctrl+Space
model = nn.Tra|

↓ IDE Intelligent Suggestions
Transformer(d_model, nhead, ...)  ← Selected
TransformerEncoder(encoder_layer, num_layers)
TransformerDecoder(decoder_layer, num_layers)

# Auto-completion with parameter hints:
model = nn.Transformer(
                       d_model=|, # int: model dimension
                       nhead=8, # attention heads
                       )

# Context-aware suggestions based on PyTorch docs
```

## Agent-Driven Interaction



*AI agents now interact directly with environments, while humans maintain oversight and provide high-level guidance.*

# Claude Code: Agent in Terminal



```
* Welcome to Claude Code!

  /help for help, /status for your current setup

  cwd: /Users/reself

> Hello World!
```

Tip 1 Install in terminal by `npm install -g @anthropic-ai/claude-code`. Login with Claude account ( 20$ per month). No extra cost.

## Demo Time

Let’s try interacting with the agent using natural language, as simple as you are a 5-year-old child.

## Memory

- **Global Memory**: `.claude/CLAUDE.md`
- **Local Memory**: `CLAUDE.md` within each project root
- **Session Context**: Maintained during active conversations

Tip 2 Use `/init` to create `CLAUDE.md` before you start.

Tip 3 Use `# <sth>` to directly add a something to the memory, or you can also let the agent to do it for you.

## Permission

### Permission Settings

```
"permissions": {
    "allow": ["Bash(ls:*)", "Bash(git status:*)", "python:*"],
    "ask": ["Bash(git commit:*)", "Bash(sbatch:*)"],
    "deny": [],
    "additionalDirectories": ["/mnt/..."]
}
```

Tip 4 Gradually refine your permission settings through daily use. **Never** set auto-accept in a single session.

## Custom Commands

- Create custom slash commands for frequent tasks
- Store them in your `.claude/commands`
- Also global and local commands are supported

**Example Custom Command `check-slurm.md` → `/check-slurm`**

```
# Check SLURM Jobs
Check my current SLURM job status and queue information.
Direct squeue command without bashrc dependencies.
## Command
squeue -u <your-id>
```

## Statusline

### Claude Code Statusline Display

CLAUDE COSTS

```
Today     $8.56
Total     $231.53
```

TODO LIST

```
increasing batch size?
```

SLURM JOBS

```
loopb4p4s4l6e4                8:17:54     H100x4
loopb4p4s4l6e4_le_iiadd_ni    QUEUE #4    H100x4
```

RECENT COMPLETED

```
loopb4p5s2l6e5                 8h ago
```

## Statusline

**Tip 5** Use `/statusline` to set the statusline.

**Homework** Configure your own statusline display.

## Image Input

- Claude Code supports image analysis and processing
- Reference images using `ctrl+v` command in keyboard

**Example Image Input**

```
> [Image #1] Please analyze this screenshot
and help me recreate the layout in CSS.
```

Tip 6 Image inputs is quite unreliable, avoid using it unless you have to.

## MCP - Model Context Protocol

**Universal Standard**: Open protocol for connecting AI systems with external tools and data sources. Pre-built servers for Google Drive, Slack, GitHub, Git, Postgres, Puppeteer, etc.

Tip 7 Use `@mentions` to reference MCP resources. What I used frequently is `@filesystem` and `@discord`.

Homework Add third-party MCP servers via `https://docs.anthropic.com/en/docs/claude-code/mcp`.

## Agents

Agent Creation Interface

```
> /agents
Create new agent
Describe what this agent should do and when it should be used (be
comprehensive for best results)

the agent will help me polish the papers by looking a latex-based
repo

Press Enter to submit  ·  Esc to go back
```

Tip 8 Use /agents to create **detailed** and **specialized** agents for recurring tasks that Claude Code automatically invokes.

## Why Codex?

- Why do we still need Codex?
- What's the difference between Codex and Claude Code?

Tip 9 Install in terminal by `npm i -g @openai/codex`. Login with OpenAI account ( 20$ per month). No extra cost.

## Claude Code vs Codex

**Claude Code**

- A much more mature product
- More like a software engineer
- Multi-step planning
- Rapid action
- Limited usage

**Codex**

- Powered by GPT-5 Thinking
- More like a researcher
- Multi-step reasoning
- Test-time scaling
- Allow more usage

Tip 10 Best practice is to use both tools complementarily.

## Setup Codex

- Launch Codex in terminal with `codex` command
- Set `\model` to `gpt-5 high` for best performance
- Configure `\approvals` to `read-only` for safety

1. **Read Only** (current) Codex can read files and answer questions. Codex requires approval to make edits, run commands, or access network
2. **Auto** Codex can read files, make edits, and run commands in the workspace. Codex requires approval to work outside the workspace or access
3. **Full Access** Codex can read files, make edits, and run commands with network access, without approval. Exercise caution

## Integrate Codex and Claude Code

Tip 11 Use `ln -s CLAUDE.md AGENTS.md` to create a symbolic link that allows Codex and Claude Code to share memory.

## Why I'm Optimistic About Codex

- **Reasoning**: Reasoning capabilities with test-time scaling by RL.
- **General Agent**: OpenAI has already built a general agent in web. *Web is the next new terminal*.
- **Multi-modal**: OpenAI cares more about multi-modal input and output.

Future Future agents will cover the vast majority of multi-modal contexts in our daily lives as prefill, and execute through extremely complex environmental interactions. Therefore, we will simultaneously have ultra-long context capabilities. How to handle conflicts between contextual information and pretrained knowledge will also become a key focus for future model development.

## Fun Fact

This entire slide deck was collaboratively created, refined, and polished using Claude Code and Codex