

Emerging Technologies

A lot of things are happening right now

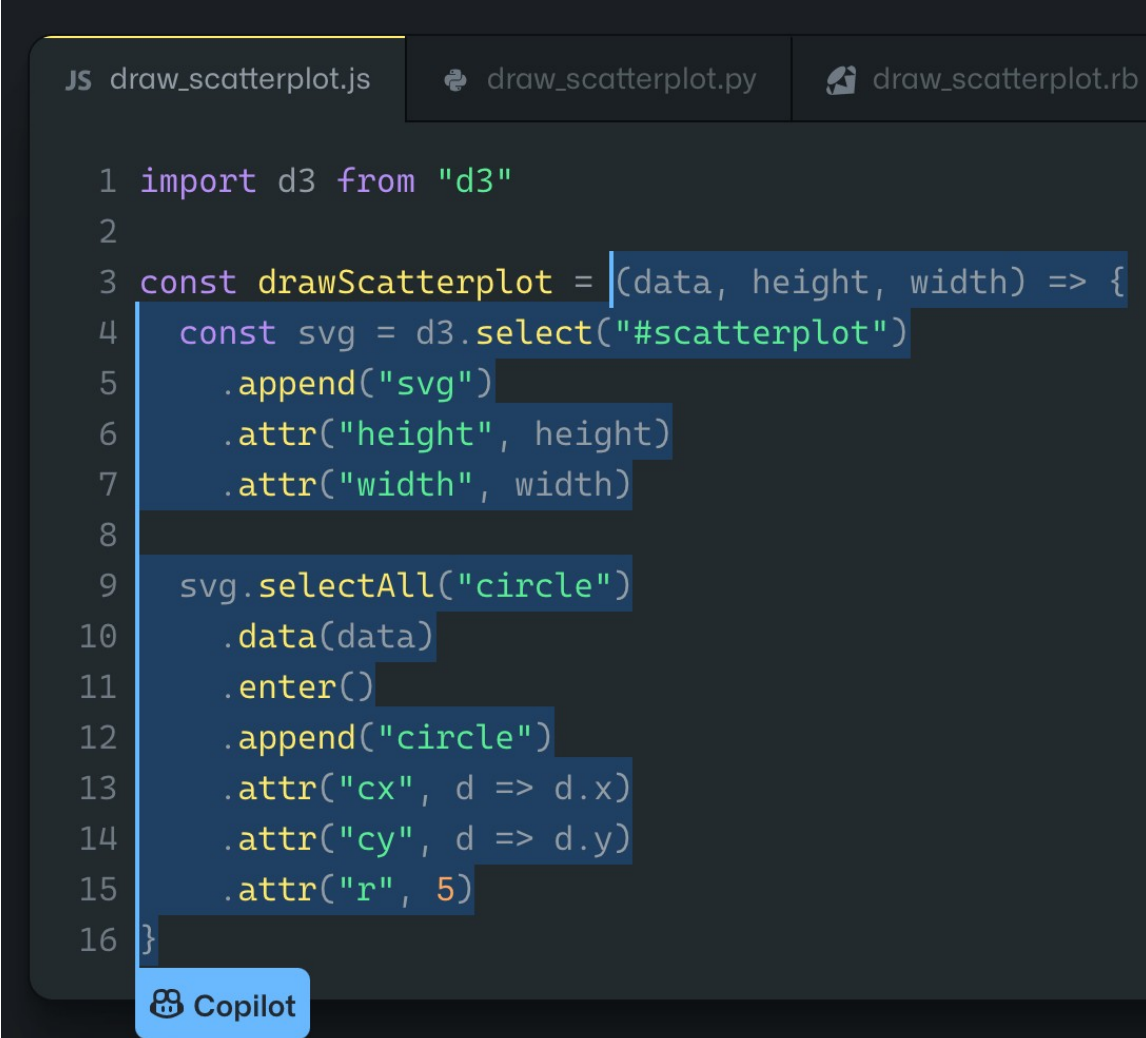
- Proof-of-work & proof-of-stake cryptocurrencies
- Web3 and decentralization
- Zero-trust
- Digital collectibles (NFTs)
- Deepfakes
- Generative AI
- AI programming assistants (Github Copilot)
- Self-serve kiosks
- The 'Metaverse'
- Virtual reality
- Augmented reality
- Wearable computing
- Internet of Things (IoT)
- No-code
- General image recognition
- Quantum computing
- Everyday robotics (food delivery robots, etc.)
- Self-driving vehicles
- Smart cities
- ...

What this lecture is and isn't.

- There's too many topics to cover, so we'll only be looking at a very narrow slice for today.
- However, in section discussions you'll be getting broader exposure to some of these topics!
- This lecture isn't meant to discourage or shame you for feeling excited about new technologies.
- This lecture is meant to encourage open discussion on their impacts so that you can treat new technologies with care.

What is GitHub Copilot?

- AI pair programming developed by Microsoft
- Generates suggestions based on your code and comments
- Trained on dataset of public Github repositories
 - Why might this be a problem?



The screenshot shows a code editor with three tabs: `JS draw_scatterplot.js`, `draw_scatterplot.py`, and `draw_scatterplot.rb`. The active tab is `JS draw_scatterplot.js`, which contains the following code:

```
1 import d3 from "d3"
2
3 const drawScatterplot = (data, height, width) => {
4   const svg = d3.select("#scatterplot")
5     .append("svg")
6     .attr("height", height)
7     .attr("width", width)
8
9   svg.selectAll("circle")
10    .data(data)
11    .enter()
12    .append("circle")
13    .attr("cx", d => d.x)
14    .attr("cy", d => d.y)
15    .attr("r", 5)
16 }
```

GitHub Copilot suggestions are shown as blue overlays on the code. A blue bar at the bottom left of the editor contains the Copilot logo and the text "Copilot".

Dataset Problems

- Lack of consent
- Plagiarism
- No attribution to copyrighted works
- Open source != free from license restrictions
 - Fair use under MIT license differs from fair use under GNU public license



Tim Davis
@DocSparse

@github copilot, with "public code" blocked, emits large chunks of my copyrighted code, with no attribution, no LGPL license. For example, the simple prompt "sparse matrix transpose, cs_" produces my cs_transpose in CSparse. My code on left, github on right. Not OK.

```
1 // sparse matrix transpose
2 CS.

#include "cs.h"
/* C = A' */
s *cs_transpose(const cs *A, csI values)

csI p, q, j, *Cp, *Cl, n, m, *Ap, *Al, *w;
double *Cx, *Ax;
CS *C;
if (!ICS_CSC(A)) return (NULL); /* check inputs */
m = A->m; n = A->n; Ap = A->Ap; Al = A->Al; Ax = A->Ax;
C = cs_spalloc(n, m, Ap[n], values && Ax, 0); /* allocate result */
w = cs_calloc(m, sizeof(csI)); /* get workspace */
if (!C || !w) return (cs_done(C, w, NULL, 0)); /* out of memory */
Cp = T->hp; Cl = T->hl; Cx = T->hx; Tl = A->xl; Tj = A->xj;
for (p = 0; p < A->m2max; p++) w[Tl[p]]++; /* column counts */
cs_cumsum(Cp, w, m); /* column pointers */
for (j = 0; j < A->n; j++)
{
    for (p = A->hp[j]; p < A->hp[j+1]; p++)
    {
        Cl[q = w[Tl[p]]++] = j; /* place A(l,j) as entry C(j,l) */
        if (Cx[q] = Ax[p]);
    }
}
return (cs_done(C, w, NULL, 1)); /* success; free w and return C */

1 Synthesizing 10/10 solutions
2
3 =====
4
5 Accept Solution
6 cs_transpose(const cs *A, int values)
7 {
8     int p, q, j, *Cp, *Cl, *w, *Tl, *Tj;
9     double *Cx, *Tx;
10     CS *T;
11     if (!ICS_CSC(A) || !ICS_TRIPLET(A)) return (NULL); /* check inputs */
12     T = cs_spalloc(A->m, A->n, A->m2max, values && A->hx, 0);
13     w = cs_calloc(A->m, sizeof(int)); /* get workspace */
14     if (!T || !w) return (cs_done(T, NULL, w, 0)); /* out of memory */
15     Cp = T->hp; Cl = T->hl; Cx = T->hx; Tl = A->xl; Tj = A->xj;
16     for (p = 0; p < A->m2max; p++) w[Tl[p]]++; /* column counts */
17     cs_cumsum(Cp, w, A->m); /* column pointers */
18     for (j = 0; j < A->n; j++)
19     {
20         for (p = A->hp[j]; p < A->hp[j+1]; p++)
21         {
22             Cl[q = w[Tl[p]]++] = j; /* place A(l,j) as entry */
23             if (Cx[q] = A->hx[p]);
24         }
25     }
26     return (cs_done(T, NULL, w, 1)); /* success; free w and re
27 }
28 // Language: cpp
29 // Compare this snippet from Untitled-1.cpp:
30 //
31 // cs *cs_transpose(const cs *A, int values)
32 // {
33 //     int p, q, j, *Cp, *Cl, *w, *Tl, *Tj;
34 //     double *Cx, *Tx;
35 //     CS *T;
36 //     if (!ICS_CSC(A) || !ICS_TRIPLET(A)) return (NULL); /* check inputs */
37 //     T = cs_spalloc(A->m, A->n, A->m2max, values && A->hx, 0);
38 //     w
39 //
40 =====
```

Who is responsible for IP infringements?

- Microsoft passes the burden of IP compliance to its users.
- Users have no way of knowing if the AI-generated code contains IP violations.
 - “Copilot completely severs the connection between its inputs (= code under various open-source licenses) and its outputs (= code algorithmically produced by Copilot)”
- Butterick claims that any Copilot-generated code “may contain lurking license or IP violations”, and the only ethical position is to not use it.
 - What do you think about this argument?

Lawsuit filed against Github Copilot

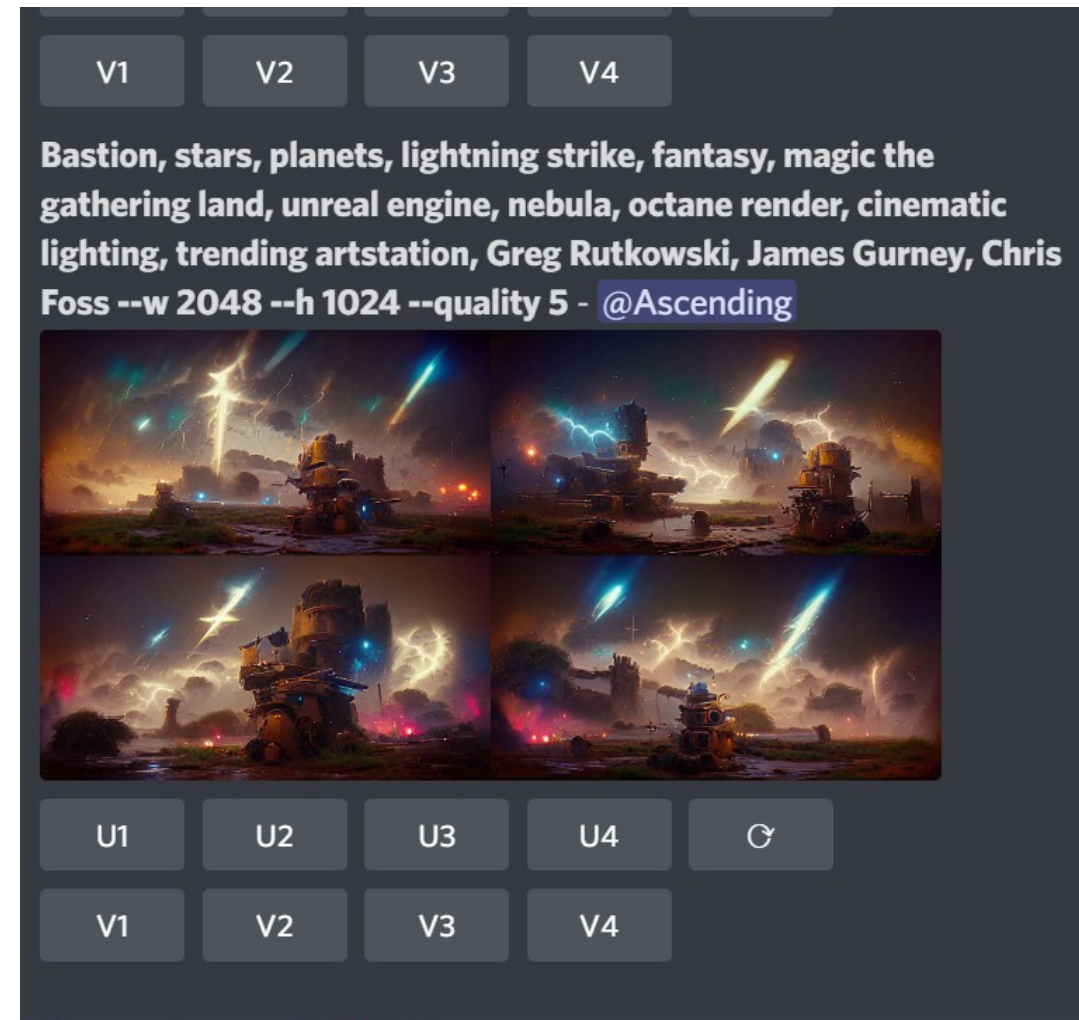
- Lawsuit filed fairly recently (November 3rd, 2022)
- Legality of Github Copilot being challenged under the following premises:
 - Training ML models on public Github repositories with no attribution violates the legal rights of creators who post code under open-source licenses.
 - Github is violating its own Terms of Service and Privacy Policies.
 - Github violates the California Consumer Privacy Act (similar to GDPR) by using personally-identifiable information in its data and failing to provide an avenue for users to alter or delete their data fed into Copilot.

Copilot in academic settings

- Many CS classes teach students algorithms at a high level and create assignments to implement them.
- How does a university deal with students asking Github Copilot to `// write a breadth-first search?`
 - What might existing ethical approaches say?

Generative Art

- Traditionally meant “computer art”, or any art that used algorithms to make.
- Now encompasses artificial learning techniques that feed in vast amounts of human-created work to produce AI-created derivatives.
 - Does this sound familiar?
- Popular programs include Midjourney, DALL-E, Stable Diffusion
- Keywords as input → Art as output



Greg Rutkowski

- Distinctive Dungeons and Dragons fantasy art style.
- Rutkowski's name is being used without his permission as a keyword to create derivative artworks based off of his brand.
 - 93,000 images on Stable Diffusion contain Rutkowski's name
 - More popular than Picasso, Leonardo da Vinci, Vincent van Gogh

Notice that the image used in the last slide had his name (among with other artists) as an input keyword.



"Dragon Cage" by Greg Rutkowski. Greg Rutkowski



Images created when Insider typed "Dragon battle with a man at night in the style of Greg Rutkowski" into Stable Diffusion. Stable Diffusion

Potential Harms to Artists

- A large part of an artist's revenue comes from image licensing and commissions from people seeking their style.
- Artist's original style may lose value
- Commercialization without consent
- "If an AI is copying an artist's style and a company can just get an image generated that's similar to a popular artist's style without actually going to artists to pay them for that work, that could become an issue."

IP for non-human work is a gray area.

- Stock image libraries like Getty Images refuse to host AI-generated art because of the legal uncertainty.
- Stable Diffusion allows their images to be used commercially, but makes no guarantees on copyrightability.
 - Decision is deferred to individual nations to figure out.
- US Copyright Office representative claims that purely AI-generated work lacks the “human authorship” necessary for copyright claims.
 - Purely AI-generated works would be rejected for copyright registration.
 - But is the act of writing keywords significant enough for human authorship?

Who benefits? Who loses?

For any technology, we need to consider...

- Who does this benefit?
- Who does this harm?
 - Who is left out?
 - Is it possible to redesign parts of this technology so that these groups are not harmed?
- What are the broader implications for society?

Prior to section this week:

- Each person should bring an example of a relatively new technology (within around the past 5 years) to discuss.
- Review the discussion guide before attending section this week!
- This is your last section! Do something nice for your TA if you can :)

Exam on Wednesday

- Covers any potential topics from second half of the semester!
 - From Software as Intellectual Property to Emerging Technologies
- Same format:
 - 50 minutes
 - In-person
 - Answer 3 out of 4 questions