

Zur Diskussion gestellt ...

AI-Generated Code Has A Staggeringly Stupid Flaw

It simply doesn't work.



[Will Lockett](#)

Jul 24, 2024

AI promises to revolutionise almost every aspect of our lives. But, one industry that appears to be particularly susceptible to AI taking over is programming. There's a virtual tsunami of articles there on how models like ChatGPT can create vast amounts of intricate and complex code from basic prompts. As such, some AI advocates are claiming programming will soon be an obsolete career, with the industry entirely replaced by AI. However, even the most sceptical AI predictions suggest that the global number of programmers will drastically shrink in the coming years as these AI models make the profession drastically more efficient. But, all of these forecasts are wrong. They don't consider a major flaw inherent to AI that it simply cannot overcome. Let me explain.

So, what is the problem with AI-generated code?

Well, one of the internet's favourite developers, Jason Thor Hall of Pirates Software fame, described it best in a recent short. He said, "We have talked to people who're using AI-generated code, and they are like, hey, it would take me about an hour to produce this code and like 15 minutes to debug. And then they are like, oh, the AI could produce it in like 1 minute, and then it would take me like 3 hours to debug it. And they are like, yeah, but it produced it really fast."

In other words, even though AI can write code way faster than a human programmer, it does such a poor job that making the code useful actually makes it far less efficient than getting a qualified human to just do the job in the first place.

But Thor's take is far from unfounded. In fact, there's a recent study that supports his opinion.

Researchers at Princeton and the University of Chicago recently found that generative AIs, such as ChatGPT, and even coding-specific generative AIs, are functionally useless.

They took 2,300 common software engineering problems from real GitHub issues, mostly debugging issues or feature requests, and evaluated how well these AIs can solve the coding issues or generate new feature code. What they found was telling. On average, only 4% of the time did AI actually generate solutions that worked, and the vast majority of these were straightforward engineering issues.

Big, dig a little deeper, and it gets so much worse. The AI model that faired the best was Claude 2, which provided a good solution 4.8% of the time. But, ChatGPT-4, by far the most complex and popular generative AI in the world and the one being used to generate the most code, only provided a good solution 1.7% of the time.

It's no wonder Thor found that using AI to code is incredibly inefficient. Imagine having to debug and re-write over 95% of the code you write.

So, why is AI like this?

Well, AI doesn't actually understand what it is doing. These generative AI models are basically overly developed predictive text programs. They use statistics based on a stupidly large pool of data to figure out what the next character or word is. As such, No AI actually 'knows' how to code. It isn't cognitively trying to solve the problem, but instead finds an output that matches the statistics of the data it has been trained on. As such, it gets it massively wrong constantly, as the AI isn't actually trying to solve the problem you think it is. As such, even when the coding problem you are asking the AI to solve is well-represented in its training data, it can still fail to generate a usable solution simply because it doesn't actually understand the laws and rules of the coding language. This issue gets even worse when you ask it to solve an AI problem it has never seen before, as the statistical models it uses simply can't be extrapolated out, causing the AI to produce absolute nonsense.

This isn't just a problem with AI-generated code but every AI product, such as self-driving cars. Moreover, this isn't a problem that can be easily solved. You can't just shove more training data into these AIs, and we are starting to hit a point of diminishing returns when it comes to AI training ([read more here](#)). So, what is the solution?

Well, when we treat AI as it actually is, a statistical model, we can have tremendous success. For example, AI structural designs, such as those in the Czinger hypercar, are incredibly efficient and effective. But it falls apart when we treat AI as a replacement for human workers. Despite its name, AI isn't intelligent, and we shouldn't treat it as such.



[Pierz Newton-John](#)

[10 days ago \(edited\)](#)

Well that's just wrong, because I and every developer I know is using generative AI to assist and accelerate their coding. It's a massive productivity boost and anyone who thinks it isn't just isn't working in the industry. To what extent it will replace workers though remains to be seen. You can't expect it to manage the whole project. It helps generate small pieces of code rapidly.

158

Hide replies

Reply



[James B Maxwell](#)

[9 days ago](#)

My use of it has evolved quite a bit over time. Mostly what's changed is that I now never assume it's produced correct, functional code, but rather think of what it's done as a kind of template. It does a lot of fast typing and comes up with a.....

Read More

43

Hide replies

Reply



[Pierz Newton-John](#)

[9 days ago](#)

Right. I made the mistake recently of asking for a function that produced Wilson confidence intervals and the not checking the results line by line, because the code looked plausible as did the outputs. It wasn't for production code, so I was in.....

Read More

17

Reply



[Mike McDowell](#)

[10 days ago](#)

Yes of course it will help the people who UNDERSTAND coding, especially when they know how to prompt it for shorter/smaller solutions.

However it won't replace developers outright. At least not anytime soon so long as AI is driven by simply predicting the next probable token.

37

Hide replies

Reply



[Pierz Newton-John](#)

[10 days ago](#)

No I agree. But Will suggests it is functionally useless since the more code it generates, the more debugging is required, cancelling out the benefit. That's simply empirically wrong, and my daily work flow proves it.

6

Reply



[Mr Nameko](#)

[6 days ago](#)

Yes. But it is only good for writing very generic helper methods or unit tests etc. What it isn't so good at, is understanding how to return code specific to the wider context of an application.

And actually that's fair enough.

It is up to the programmer to give AI, a good enough prompt. Now, it would take a hugely detailed prompt to build a method, that understands how to integrate into the wider application.

The programmer would be better off just creating the code himself, than creating the prompt for this code.

So, I agree that AI will be a good tool for helping programmers, but I cannot see it replacing programmers.

7

Reply



[Steven Lueck](#)

[9 days ago](#)

I recommend that Will should check out "Writing an Arduino driver with OpenAI ChatGPT and PDF parsing." youtube.

com/@adafruit. This will give you an example of proper use of AI. If you are just playing with AI that's all well and good, but don't expect it to do much beyond entertainment value. If you have a specific problem to address and take the time to train your model then it can be very useful.

2

Reply



[Graeme Smith](#)

[5 days ago](#)

Very much this. I use ChatGPT and Perplexity all the time to write small snippets of code when I can't remember the exact syntax, or to convert code from one language to another. Copilot I have a weird relationship with. For python code it often feels like it is reading my mind with it's predictions of what I want to write, whereas in C# it seems completely clueless and is more of a hindrance to productivity continually trying to suggest 20 lines of code that has nothing to do with what I am trying to write.

1

Reply



[Dan O](#)

[10 days ago](#)

Will you are all wet here.

I know you are not a SW dev, you are a journalist. Still you need to do your homework.

MILLIONS of devs are using LLMs to speed their work every day (including myself).

It just WORKS. As an AI guy I am shocked how well it works.

Does it code correctly out of the box. No. Is that relevant here. No!

What matters is how fast I am with and without it.

I do need to understand the code, because as you say the LLM really doesn't.

BUT it gets hundreds of details of API signatures correct etc. I just need to ensure the overall approach it is taking is a good one, and occasionally I need to debug when it cannot.

But even in this area >80% of the time, I cut-paste the error message and ask it what is going wrong, and it usefully points me in good directions and suggests fixes to its own code.

Your referenced study is irrelevant, and your conclusions are simply at dramatic odds with many many programmers who are coding right now.

91

Hide replies

Reply



[Adam Puckey](#)

[he/him](#)

[1 day ago](#)

Funnily enough when I read 'it takes me 3 hours to debug' I did wonder if that time would decrease with practice. Have you found that to be the case?

Hide replies

Reply



[Dan O](#)

[1 day ago](#)

Hey Adam,

I don't find the 3-hour debug to be the case.

Indeed the first thing I do when I get an error is to cut paste my call and error traceback and ask chat GPT how do I fix this? It remembers the code and provides the fix!

BUT:

-- it does take time to learn how to prompt GPT in a good way to get the result you desire.

-- and I am a seasoned programmer, so I usually know what I want and I really tell GPT. hey lets use a "dict of dict of dict of set of observations, indexed by ..." (this is an actual example from yesterday)

-- AND it really does understand what is going on, so it is important at a high level and low level you DO. So when it proposes an approach that is just not a good one, you can either reframe your ask, or give up on GPT for that example.

It might sound like I am doing all the work when I get in there and tell GPT an exact data structure or method signature, but not really. because I just tell it bits and pieces to get it going in the direction I want, then let it fill in 90% of the detail.

With those caveats in place I am 2x to 5x faster for certain frequent tasks, and I am learning which tasks those are. Right now I very much am sill the programmer, just with a stupidly powerful "auto-complete"

Reply



[stephen lilley](#)

[10 days ago](#)

In the vast landscape of battles you could wage against AI, this is one you'll never win. The moment you challenged it, AI had already 10 steps ahead. Just yesterday, Meta unveiled their latest breakthrough, the 405b model. It's as if we're trying to rein in a horse that's already bolt and left the property. Debating its merits is futile when the finish line keeps disappearing into the distance.

I'm no seasoned developer—far from it—but in just a few months, AI has enabled me to achieve feats in C and C++ that I once thought impossible. The speed at which a novice like me can iterate through code to find a working solution is nothing short of miraculous. Sure, it might not match the finesse of a veteran coder, but for someone with little to no coding background, it's a 100x boost in capability.

AI doesn't tire, it doesn't scoff, and it's never too busy to help. Its willingness to teach and advise is unwavering. When it comes to probabilities, the chances of AI getting the code right are exponentially higher than mine with zero-shot prompting. The future isn't just coming—it's already here.

51

Reply



[Thomas Møller Jørgensen](#)

[11 days ago](#)

A short, and kind of parallel, story from real life:

My father once worked in a cleaning company. One day a salesman arrived with a new series of products. After delivering his pitch about how great these products were, my father asked him some questions. One of these was:

This new soap, what is its pH value (not an unreasonable question when you work in the cleaning business).

The salesman didn't even blink: It is zero!

Which of course would mean that it was so acid as to be quite dangerous to work with.

The point is: The salesman knew the form of a positive answer, but had no idea what he was talking about .

That it how our current AI works. And for some things the form is good enough. For things such as code, it is not.

95

Reply



[David D](#)

[11 days ago \(edited\)](#)

I wish we would stop calling LLMs AI

104

Hide replies

Reply



[Evan Loughlin](#)

[3 days ago](#)

Alas, the AI Effect strikes again...

https://en.wikipedia.org/wiki/AI_effect

1

Hide replies

Reply



[David D](#)

[3 days ago \(edited\)](#)

There are genuine AIs out there, LLMs are just not one of them. They are impressive but, like a chess program, they are not 'intelligent'. It will not come up with anything 'new' unlike for example those google AIs that created new programming languages on their own. Or AI that can develop new drugs that people never dreamed of.

Or maybe my definition of Intelligent doesn't match what an AI marketing team puts out.

9

Reply



[Turner Rentz, III](#)

[11 days ago](#)

Disagree. AI generates snippets well, that's all people are really using it for right now. Nobody is generating full stack.

A full 60 percent of developers state that AI helps improve their ability to write code and that they're using it.

22

Hide replies

Reply



[Mr Nameko](#)

[6 days ago](#)

No. You have completely misunderstood what the author is saying. He is saying that AI won't replace the programmer. It is great for assisting a programmer, but it will never build bespoke applications, by itself. And the reason is actually very simple. AI is only as good as the accuracy of the prompt. Imagine building a prompt that would build an entire application made up of 1000s of class components that are injected into one another, with database connections etc. The prompt would have to be massively detailed. You might as well write the code than write the prompt. What AI is a great at, is creating utility methods, unit tests and general introspection. It is there to help the programmer and not replace us.

63

Hide replies

Reply



[Bill Coop](#)

[2 days ago](#)

I have just written a text recognition program in C. The bones of the program is a page long, and basically a single idea which could not be broken down into bits. When I posted an explanation I must have pressed cancel, as it disappeared.....

Read More

Show more replies



[Andy Forest](#)

[11 days ago](#)

Sigh. AI models are getting better all the time. Your article is a long version of 'AI isn't perfect'

37

Hide replies

Reply



[Jim Cross](#)

[11 days ago](#)

It's a little more than than AI isn't perfect. The problem discussed here is a problem of AI as statistical model. With that approach, it is has limitations in what it can do. Adding more training data won't fix it.

36

Reply



[Jeff Heisler](#)

[10 days ago](#)

A year ago I started using AI to generate Flutter and Dart code. I started with just some small classes, functions, maybe some models. All guided deeply.

Chat worked great, not so much any of the other AIs. Last month, using 4o, "I" built a small but not inconsequential app, <http://cm2git.web.app/>, that is basically a Kanban with a variable number of columns, along with limits... Kanban Cards... A github repo on the back to track cards with... Lots of charts and graphs. It's pretty complete.

80% of the code was written by AI. Maybe more. It's gotten remarkably better. As an older dev (66, 40 years as a dev), Chat 4o has become the assistant I can't afford.

Will it replace me? NO. Will it assist me? Absolutely!

16

Reply



[Rex Kerr](#)

[10 days ago](#)

As others have said, the premise of the test is wrong. Of course if you use tools for tasks they're not suited for, the results will be poor.

If you use AI to help build small well-described subsets of the problem, which you fully understand, you can considerably reduce development time.

If you ask for too much, and get nonworking garbage, you shouldn't spend hours debugging it--you throw that away, and design a solution yourself, then ask the AI to do smaller pieces. Keep doing it until you get into the sweet spot where AI does it right (or you realize that it's better to do it yourself). Shifting your work to more software design (which you always had to do) and less line-by-line coding is where AI provides its benefits.

17

Reply



[Cam Cairns](#)

[11 days ago](#)

AI has some utility in developing code with very complex syntax and lots of boilerplate.

Rust, for example, is notorious. Really helps to have AI generate delights like:

```
fn on_event(&self, e: &Event) -> Option<Pin<Box<dyn Future<Output = ()>>>> {
```

```
...
```

```
}
```

33

Reply



[Del Baker](#)

[9 days ago](#)

Perhaps Artificial Incompetence would be a more appropriate use of the AI acronym? ;).

67

Reply



[Tim Graham](#)

[12 days ago](#)

it can definitely accelerate my work but doesn't replace it.

20

Reply



[Disruptive Concepts](#)

[10 days ago](#)

The AI model that faired the best was Claude 2, which provided a good solution 4.8% of the time.

Claude 2 being the top performer at a staggering 4.8% is like celebrating the best house on a street full of fixer-uppers. The bar is low, folks, and we're just stumbling over it.

#TechReality #AIFailures

25

Reply

N T
T P

[NTP](#)

[12 days ago](#)

Wow, some AI investors are about to get a rude awakening called reality.

24

Hide replies

Reply



[Brindlephilip](#)

[11 days ago](#)

If you listen to somebody like Will, the author, good for you...

1

Hide replies

Reply

N T
T P

[NTP](#)

[11 days ago](#)

He is good, right? I try to keep up w/ his stuff!

17

Reply



[Shawn Murphy](#)

[12 days ago](#)

ChatGPT four couldn't tell the difference between SWQL and SQL in my recent attempt at use. When prompted to understand that there is a difference, it still produced an answer in SQL instead of SWQL. Then when told again it was wrong, it apologized and finally produced a piece of code that was usable as SWQL. It wasn't as granular as what we were already using. It's Solarwinds port of SQL but the documentation is readily available, and doesn't change much, although occasionally. I have had it at least direct in the right way with some nested loops in C, though.

10

Hide replies

Reply



[Henry Kaye](#)

[2 days ago](#)

That's because they dropped the Machine Learning bit out of AI after the initial phase. I have experienced those "apologies" time after time, in both ChatGPT and Gemini, and in generating text and code. The simple fact is that it shouldn't be called artificial intelligence. It's artificial extrapolation. That's all it does, it extrapolates a random answer from existing knowledge. It neither learns nor does it improve the data from which it extrapolates, so it actually makes the same mistakes over and over.

14

Reply



[Milko Škofič](#)

[7 days ago](#)

The bottom line is that if you need the UI response to be correct, you must still check it, there is no guarantee that it will produce code without bugs. On the other hand, most of the time it shows you a path which helps if you are stuck.

Would you trust AI to program a machine that keeps your vital organs working?

12

Reply



[Steve GG](#)

[8 days ago](#)

You use the term correctly only a few times, ie AI != Generative AI = LLM. Please get it straight, otherwise you sound like a fool.

You completely ignore Copilot which is one of the most used AI tools in coding. Copilot cannot write a feature, but it can complete a script or write unit tests or complete mechanical type code that enumerates similar things.

14

Hide replies

Reply



[rogue4gay](#)

[8 days ago](#)

Have used copilot. Great for generating appropriate templates!

Hide replies

Reply



[Steve GG](#)

[8 days ago](#)

Exactly. Only an idiot would use ChatGPT or worse, Gemini, to write functions for them...those same people are deploying production releases to CrowdStrike.

13

Reply



[Matt Friedman](#)

[4 days ago](#)

I challenge the premise. If you say the LLM must produce the right result in one shot then yes the models can't do so. But that's not how programming works. Ai is an assist that makes a developers time more valuable. As such it's not a replacement and may never be so.

6

Reply



[Caleb Howard](#)

[6 days ago \(edited\)](#)

I'm easily 10 times as fast a coder since using LLMs.

This article shows a lack of understanding of the vast utility - that's all.

Write about what you know, or enlist folks who do know to vet what you write, maybe.

It is sufficient to say that generative models have already changed the world permanently, and completely - and they are yet in their absolute infancy.

Hang on to your hat, Mr. Lockett.

1

Reply



[Vicduoba](#)

[6 days ago](#)

But the test used code that had bugs. Not an unbiased test of how frequently AI produces code that works. I find that you do have to phrase your initial request carefully and then examine the answer, using intelligently-phrased correctional dialogues. I find AI very productive for SQL coding and I don't give a hoot about the nay-sayers. You have to learn about the AI positives, not gloat over the negatives.

1

Reply



[Ernest Tan](#)

[7 days ago](#)

As long as we know the limitations of this tool, it still can be useful. LLM's weakness is it is just a sophisticated parrot with zero understanding. Whereas humans can understand and test out the code effectively. Some bugs generated by generative AI are hard to spot though.

9

Reply



[Juan Chulilla, PhD](#)

[8 days ago](#)

Excuse me, are you from the past?

1

Reply



[Nickolas Pitfield](#)

[8 days ago](#)

1 - garbage in, garbage out

2 - AI isn't actual intelligent - it cannot reason, it has no intuition or gut feel, it has no sense of right or wrong - it's just the sum of some thoughts of a small set of providers, with all their biases and limitations.

10

Reply



[gravity well \(Rob Tomlin\)](#)

[9 days ago](#)

I disagree with your conclusion. A lot depends on how you prompt. I find it quite helpful. Even complex Snowflake queries.

2

Reply



[David Byrne](#)

[9 days ago](#)

I don't *rely* on "AI" to code, though when I'm feeling lazy I'll often ask it to spin up a few hundred lines to save me typing 😊 I can read it and tweak it if necessary, as easy as proofreading a letter in English, but saves me typing 😊

If I don't know exactly what I need to code it's easier for me to work it out myself rather than try to explain something I don't yet understand: if I can't explain the problem in code, then explaining it as a prompt won't fix my lack of understanding.

However, as an example, yesterday I spent the morning searching for a bug in a piece of code (1000-ish lines), without success. I pasted half of it into chatgpt with the prompt "what's wrong with this?" and the answer was in the first line of its response. That would have improved my productivity yesterday by 50% ... 😊

3

Reply



[Olaldennis](#)

[9 days ago \(edited\)](#)

Umm...I think you haven't **really** tried Claude 3.5 yet.

3

Reply



[Auntiegrav](#)

[10 days ago](#)

Though I am almost always going to assume the robot is less wrong, In this case, it's a lot less wrong because the problem is like someone trying to use a pair of pliers to remove lug nuts. The people who know how to use the tool have already moved on to the next task (and don't answer surveys or get grant money to 'evaluate') while those expecting miracles are scuffling around looking for their lost fingers. I like the story because it brings great discussion from techs.

4

Reply



[Russell Salsbury](#)

[12 days ago](#)

The is no I in AI because it doesn't understand. LLM has no mechanism to understand. With very constrained datasets and domains, an AI generator can produce amazing results, but as a general tool its a dead end.

I can see the solution, but this comment is too small to contain it.

4

Reply



[Pascal Meheut](#)

[about 22 hours ago](#)

The article is based on a quote and a study whose premises are never challenged.

IA generated code is like every other tool: very efficient if you know how to use it and its limits.

Of course, if your problem is a screw and you use it as a hammer, you can "prove" that it does not work.

Reply



[Steve Vollum](#)

[about 23 hours ago](#)

very insightful. Thanks.

Reply



[Nuked](#)

[1 day ago](#)

It is like criticizing a knife because it cannot cut a cake by itself...

Reply



[Adam Puckey](#)

[he/him](#)

[1 day ago](#)

It simply doesn't work

You had me at this awesome strapline 😂

Reply



[Organizational Jazz](#)

[2 days ago \(edited\)](#)

"My use of it has evolved quite a bit over time. Mostly what's changed is that I now never assume it's produced correct, functional code, but rather think of what it's done as a kind of template."

I think this is the right way to proceed. I, too, have tried approaches ranging from prosy to as prescriptive as I can make it. At some point, you reach the limits of what statistically or hyper spatially has been "seen before", and that is where things don't quite work as well as desired.

I will agree also with those who say AI is an accelerator. In their statement and yours, there is agreement... in some cases you actually DO get code that works (there is nothing new under the sun, and this shows intersection with statistical significance in a training corpus). In other cases, not so much. In those cases, your idea of template might work but, again, maybe not so much.

BUT that doesn't mean what is being attempted is bad (and I am not saying that you or any of your readers are saying so). We just haven't yet found the little incremental nudges to optimize what we get out of these statistical marvels.

Reply



[Bill Coop](#)

[2 days ago \(edited\)](#)

I am surprised to see how positive some people are towards LLM's, and think this is because they are talking about two quite different things.

I recently wrote a text recognition program in C and found the language models were of no use, and just copied old code from others, but did not help at all. The bones of the program is a page long, with three pages joining it together; so they were a waste of time. Even on a page of code there is no way this single idea could be broken down into various parts.

I have only basic maths skills relating to electronics, so no real understanding of data science, but this seems to be where the divide is. If said program was part of a larger scheme then I'm sure Wilson data models would be useful, but otherwise not.

This is probably why data scientists like chatbots, while everybody else regards them as a glorified, but very stupid Elisa program.

Reply



[Luc Ponsaerts](#)

[6 days ago](#)

I am learning Python and C# not by following a training program, but by writing small useful apps. In that case AI is a serious help (I have to admit I was an on/off programmer since 1971, writing machine code, fortran, pascal, vb).

I have an ongoing heated discussion with an old friend , my age, English teacher, about AI. Me telling him that the I in AI is wrong and misleading people by laking them suppose that the thing is intelligent, and blown as such out of proportion by the lain press, and him, insisting on the term “artificial”, meaning that AI is not intelligent.

Reply



[Twillemijn](#)

[7 days ago](#)

Nice read!

I agree with the fact that AI is definitely not going to replace developers as a whole. Because the thinking part is super important and AI is very bad at that. It is a really good tool to speed up a developers process, such as getting quick answers to their super specific questions without always having to look through 20 stackoverflow posts. Although from experience as a developer, this is still often needed.

I think where a lot of the current ideas with AI are going wrong is in the idea that AI can do rule based work. It cannot and will never be able to do pure and consistent rule based work. It often gives correct answers on rule based problems because the answer can be generated through the training data but the LLM is not "thinking" itself.

What I think is even the extended problem with this, is that people don't recognize what is and isn't rule based. The generation of text is going so well, but then they want this generated text to be in a super specific format using super specific rules. And this is where it becomes complicated and from my perspective often inconsistent. This is a very frustrating proces but it is super interesting to see and learn through the current GenAI hype what eventually will and will not be good use cases for LLMs.

Reply



[Steven Roosa](#)

[7 days ago](#)

GPT 4 is a tremendous coding assistant. Common knowledge at this point.

Reply



[Javier Romera](#)

[7 days ago](#)

AI is for solving a certain problem.

Generative AI solves the problem of the next word, not coding or anything else. Unless you create a model that really solves that problem, not just regurgitating words after the previous words. It looks good and anybetimes works. But this is not their strength or the problem they can solve.

Reply



[Javier Romera](#)

[7 days ago](#)

AI code is very good for hobby programmers, like me. It needs a couple of interactions to make it work and sometimes you need to give away and start again. But it is ok, because many times I do not have the knowledge to do it at all, or do it in short time. I am used to its mistakes and own ideas (like ignoring utf-8 until you ask every time or giving code that does not execute at all).

But I do not see how a professional programmer would use it for other than boiler plate generation for repetitive or simple things.

Reply



[rogue4gay](#)

[8 days ago](#)

AI will be able to generate code when I can input test cases and have it create the appropriate code.

I don't believe the AI is trained on a test case to code structure.

Testing and debugging is the hardest part of writing code. AI is good at generating code that has the correct syntax. Not good at developing code that has the right functionality. How could it be if I can't input test cases.

Interested to see where Microsoft heads on the test case to code as part of co-pilot.

Reply



[Juice64](#)

[8 days ago](#)

For AI to be relevant in the coding industry it will have to understand the analog. This can only be done at this time through a neural network (biological). If non-biological neural models using mathematics are employed then more advanced electronics will have to come into play, digital technology is limited, these developments are still down the road, but achievable no doubt.

Reply



[Anil Pal](#)

[9 days ago](#)

The article on AI-generated code flaws is eye-opening! For reliable and intelligent AI solutions that mitigate such issues, visit CSimplifyIT.com. Our expertise ensures robust and error-free results.