Opus 4.8 Scored 81. Your Workflow Doesn't Care.

26 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=Z73YUF14UDI
https://www.youtube.com/watch?v=z73yuF14udI

SUMMARY

Opus 4.8 has been released amid high expectations, but it serves more as a placeholder than a groundbreaking advancement in AI models. While it shows improvements in certain tasks, it lacks the consistency and usability needed to be a daily driver for many users, especially when compared to OpenAI's 5.5 model. The discussion highlights the importance of harnesses—how models are integrated into workflows—over the models themselves.

- *Opus 4.8 is not the anticipated major release (Mythos) but a strategic release coinciding with a funding announcement.*
- *The model shows progress in long-running tasks but is not as reliable or effective as its predecessor, Opus 4.7, in practical applications.*
- *Users report that 4.8's unpredictable reasoning and overthinking hinder its effectiveness as a daily driver.*
- *The importance of harnesses is emphasized; the integration and usability of a model can significantly impact productivity.*
- *OpenAI's 5.5 model currently outperforms 4.8 in practical tasks, showcasing better reliability and efficiency.*
- *New features like SLworkflows in 4.8 show potential but require careful consideration of how they fit into larger productivity systems.*
- *The competitive landscape suggests ongoing innovation, with expectations for more powerful models and open-source options in the near future.*
- *Leaders should architect their systems for flexibility, allowing for easy integration of various models based on specific outcomes and needs.*

Everyone is getting the Opus 4.8 story wrong. And I think it makes sense that we're getting it wrong because we're used to the 2025 story. The 2025 story of AI was basically new model drops, open AI drops, cloud drops, etc. And you get a new high bar and then we talk about what that enables, what that unlocks, etc. We are in a different stage of the race and it was never more clear than when 4.8 dropped on Thursday, May 28th. What happened was this opus 4.8 in some ways by some measures is the strongest model out there right now. But that doesn't mean anymore that it's the best model or the most useful model for you. And I want to unpack what's really going on there, where I think it adds a ton of value, where it doesn't, some of the nuances people aren't talking about enough that I think indicate where big models are going, and really talk about the state of the race between the two major players left, open AI and anthropic. So the first thing to understand is that 4.8 8 is not the big model drop that everyone is waiting for from Anthropic. And we just have to get the elephant out of the room. Everyone's waiting for Mythos, right? Everyone is excited about Mythos. Mythos is the most teased model release in history. The real reason is because they had a funding announcement that they needed to announce that day and they wanted to do very classically a we release a new model that's a leader in many things

plus we just raised a bunch of money.

That is why they released 4.8 when they did. That was the reason for the calendar timing. it is not because they have the best new model out there to drop. And I think that that distinction came through in the test results. So what you see is that 4.8 has made real progress in some of these longer running tasks that we know the model makers are obsessed with because they burn tokens, right? Like these longer running agentic tasks. It does that well. It is better at paying attention and staying on task than 4.71, which is a weakness that I noticed with 4.7.

That's great. But it is not the monster intelligence super model like mythos that everyone has been waiting for and hoping for from them and that we all suspect they cannot release because they don't have the compute to deliver. And so I think the right way to look at 4.8 is it's kind of like a placeholder release. You need the release for your funding announcement. You now have a valuation close to a trillion dollars. You've raised a lot of money and you need to show that you are still in the race and you release a strong checkpoint of a model that shows that, right? That shows that you're that you're making progress and continuing to deliver and everyone's still going to be waiting on mythos. But even that, even that is not the real story, folks. Because the real story is that no matter how good it is, it is not becoming my daily driver. And it's not becoming a lot of folks daily driver because of two key differences that I think highlight some of the weird dynamics we're getting into in 2026 with very large models. And I want to be open and honest about this because I think that the way model releases and model development is working in 2026, it's really different from 2025. The first big piece that I think is preventing this from being a daily driver for a lot of folks is it does not work predictably when you scale up reasoning effort. And so for I don't know over a year now we have been told you scale up reasoning effort and you get better results.

That's what everyone says. It might be more expensive but you get the better results. That appears to not be the case predictably with 4.8. There are some situations where scaling up the reasoning level to what they call max is going to be the best choice for you. There are some situations where high will be a better choice for you. And that's super confusing because high is less than max. And that's obviously less than than the rest of the the reasoning scale that we run into with OpenAI models. Right now, if I scale up OpenAI to the extra high reasoning mode, it works better. It works predictably better. That's nice for me because it's it's really a product choice at that point. I can understand how it works.

And it's not just me saying this based on vibes, by the way. There are hard test results that show this. Vending Bench came out. Vending Bench is famously the the benchmark that shows how AI does at running an actual business which is a vending machine. Opus 4.7 did really well on this. Opus 4.8 did worse. It did worse people than 4.7. It was a regression. And that is true whether you were on high or on max for your thinking mode. And what's really interesting is that 4.8 on high beat 4.8 8 on mats for vending bench. In other words, if you are running a vending machine right now with your AI, which I don't know a lot of people doing in practice, but it's still a good benchmark. I love the focus on practical business, then you should be using 4.7.

It beats everything else out of the water. And if you use 4.8, you should use the dumber version of 4.8 because max is not good. And I think that this gets at one of the larger issues or challenges with this current direction

from the anthropic team that I want us to talk about more openly. 4.8 is a model that thinks a lot about whether something is aligned. And in principle, you want more powerful models to be aligned. I I get that. But if a model overthinks, it may become less effective. This ties in, by the way, with one of the other big beats of the last week or so, which is, of course, one of anthropic's co-founders in the Vatican in Rome when the Pope released his encyclical about AI. Effectively, the Pope picked a side with anthropic and said, "These are the guys who are thinking philosophically in a way that is aligned to where I'm going."

Anthropic spends a lot of time thinking about how to get AI right. And there is a lot I admire about that. I have so much respect for the work of Amanda Ascal and the work she's done on the constitution for Claude like it's personable that feels like it has an understanding of what it is to be human. Now, I'm not saying it does, right? I'm just saying that there is an ability to grow a model because they're grown.

They're not they're not made that feels like it understands the humanities and that comes out in things like front-end taste, which is really fuzzy, or the ability to write sentences that don't feel robotic, like all of that stuff. But you can take that too far. And it looks to me like 4.8 took it a little bit too far because if you get into a situation, and we have seen reasoning traces that are coming out of 4.8 8. Now on max mode, it looks very much like the model overthinks itself and overthinks itself specifically around the constitutional questions. And what I mean by that is what is right to say, how do I align with my constitution, etc. I I have seen reasoning traces where where you just say something fairly simple and you pull the reasoning trace for 4.8 max and it's talking about how it needs to write warm paragraphs.

It needs to align to its larger constitutional questions. needs to be aware of Amanda Ascll and sort of her preferences, which is kind of funny because I doubt that she would wish that her particular preferences are recorded in the model's thought patterns. It's more that she happens to be a fairly well-known personality now. She has done, you know, larger conversations on the internet about how she's shaped Claude. And I think that there may be some leakage from some of her public statements on Claude into the model at this point. We'll see. That's a that's a suspicion. But regardless of how you read it, 4.8 thinks to itself a lot to the point where it is less effective.

And I think that that's something that even if that's not true across the board, because don't get me wrong, you might think that I'm saying 4.8 isn't good at things. No, it's very good at some things, and I will get to that in a second. It's that it's hard to predict and reliably use it as a daily driver if you have an overthinking problem. If if it unpredictably overthinks about things, can you trust it to be your daily driver? And this brings me to the second point that I want to make. Daily drivers are increasingly a function of harnesses. I talk a lot about harnesses.

I want to make it really simple and clear. The harness is the shape of the product around the model that allows you to do useful things with the model. And I'm going to be very specific and very clear. In contrast, 5.5 in codeex with 4.8 in co-work and 4.8 in cloud code. One of the things that enabled the breakout in January is that claude code was so ergonomic for developers. You could type in plain text in the shell in the terminal. You could get what you wanted done. Claude would just understand. Increasingly, it got into sub aents. We got into the

Ralph looping. So, it would go on and do big things. It's amazing how quickly things change because that world that felt so ergonomic in January and February hasn't really changed. It's still what it was.

It's still beautiful in many ways. We have of course continued development and continued iteration from the other player in the game from OpenAI. If you are trying to tackle complex, difficult work that is on the edge of model capability, which is much farther along than it was in January, I cannot underline enough how much has moved since January, how big the agent jump has become. I know it does not show up in the Excelss. I know it does not show up in the PowerPoints. I know when I say you can do much more with agents, people roll their eyes and say, "Well, how is that going to help my Tuesday?" But the difference for anyone who is driving these models at the edge is absolutely stunning. And I I think it's really important to be honest about that because it shapes the ergonomics of our workspaces which comes back to the harness piece. Right? If we are looking at harnesses and how they are useful in late May, in June of 2026, we have to recognize that the tasks we are giving our models don't need a Ralph loop anymore because the models just know to keep going till they're done.

They don't need special help to stay on task. They increasingly don't need special help to review their work because they have been trained by the model makers who care about doing longunning tasks well to get that piece done. And so a lot of the things that we associated with the harness, they've evolved really significantly. And so when we compare 4.8 versus 5.5 today, we have to compare them for the task that we're tackling them on. And honestly, 5.5 in codec is a much much stronger harness right now, regardless of the model inside, than 4.8. And I know that in part because I've played with 4.8 and 5.5 in their respective harnesses. And I can see places where 4.8 has more insight, where 4.8 has more front-end taste, where 4.8 is the better writer out of the box. And I still find myself going back to 5.5 in codeex. And the the honest reason is the harness. And I'm going to name specific aspects of this harness. And I'm going to name them in part because I want more competition in the space. I'm not choosing OpenAI because I am picking a favorite. I am choosing OpenAI because behaviorally that is what is working for me right now. I think one of the key aspects to a harness if you are going to do these big long running tasks and for perspective I'm running multiple two three four five six hour tasks a day where I give the model a big goal and it just goes and does it there is no comparison right now between where open AAI is at and where anthropic is at on this and it's a big big gap if you do big tasks and if you're wondering Nate what are your big tasks what how are you doing this I'll give you an example Yesterday, as part of my workup of 4.8 and 5.5, I gave both of them the task to design end to end and build a website for a MD Markdown domain that I happen to own. And I was like, you should be able to just do this, right? I shouldn't have to remind you. I shouldn't have to give you verifier steps. It shouldn't just be a one-page website. I you should be able to just do this. And what I found was because of compute availability 4.8 8 just errored out and couldn't do more than than one task at a time and took forever doing it. And what I found with 5.5 is I could build two of those sites at the same time. They built relatively quickly. I did not love them, but I had time to go back and I went to chat GPT images mode and I actually had said, "Look, I'm not happy with your design initially, 5.5. I'm gonna ask chatgbt images mode to design me a better like JPEG frankly like a little PNG image that shows the front page of the site that doesn't suck. It is like welldesigned and then I'm going to feed that back to you and say look at this image make it better. And I was able to get through two websites and that iteration loop that got to a nice fully deployed on domain DNS name servers assigned complete website with 5.5 twice in the time it took for 4.8 8 to error out twice. And I just I can't help with that, right? And then there's smaller things, right? If I tell

5.5, go look in my files for X or Y, it just does it. It knows my whole computer. It can sort it out. If I tell 4.8 in the desktop app in Mac to please go look at my files, it's like, oh, I can only see downloads. I can see desktop. That's it. and it doesn't take the initiative to say, "Can I get your permission to go look at these other files since you clearly want me to?" These are the little things that make it hard to do big longunning tasks if you are an AI builder in 2026. I know it feels crazy, especially if you're a CTO or a CIO where you're like, "Oh my god, we we just did this. We just signed the anthropic contract. I know people who did this, right?" And they're just like tearing their hair out. And I am not saying that's a bad thing. One of the things I think is really important to recognize is that if you try to tie an enormous amount of your budget to one horse in this race, you are not setting your company up for success. You should be tying your budget to outcomes that you want to drive. And you should be allocating your budget against the models that work best for those outcomes. It's really very simple. And so you should be in a position with your harnesses, with your model where you can swap them out when they don't work. And it's just an API swap and that's it. and you're done. But I really want to encourage folks, don't pick a winner.

I'm not picking a winner. I'm not saying OpenAI is always going to work best. I don't know that. I don't assume that. In fact, history suggests the opposite. History suggests a continued horse race. And soon I'll be talking about how incredible Claude is because Mythos is finally released or something like that. And I'm open to that. I'm excited for that. I like the story here. I like the competition. I think it's good for all of us. But right now, 4.8 8 is reading like a checkpoint release that overthinks itself that is unpredictable and I think that illustration with 5.5 in the harness shows a lot of those aspects right it like 5.5 was able to get the files 5.5 was able to get the entire job done twice in the time it took 4.8 to think think and error out.

And these things matter. If I'm able to run two or three or four or five or up to 10 threads at once with 5.5 in harness and I error out on one or two with Claude if they're big tasks, I'm sorry. It doesn't matter how smart your model is. I can't pick it. But I said I would tell you some of the things 4.8 is good at. And I want to call one out here that I think is really important.

workflows command is a really interesting command in cloud code that came out with the 4.8 release and I want to name it because I think it shows us an interesting direction for agents in 2026 and I think it's going to get copied because sometimes these anthropic innovations they just get copied. We had a problem when we were running workflows where you either deterministically tell Claude this is the workflow I want you to run. These are the sub agents I want you to run. Uh or you let Claude sort of decide how to get the job done. But you don't get visibility. There's an in between state that SLworkflows invokes that's really interesting.

Slashworkflows as a command in claude code lets you say please compose a workflow and then claude 4.8 8 will think through the problem, compose a workflow with multiple agents, disclose that workflow, and then give those sub agents tasks in line with that dynamic workflow. It gives you transparency. It lets you see how agents are going to tackle tasks that get the whole job done. I think it's a pattern we will see copied for a lot of individual productivity agents in the summertime of 2026 because it just makes sense. Like even with codeex, I can't do that right now. That is a unique thing. I'm sure it will get copied soon, but for now, it's a unique thing with Opus 4.8. I think it's a great innovation. Hats off to the team. I love the idea. But this brings me to one of

the larger challenges in covering models in 2026. And it's true of 4.8. It will probably be true of every model going forward. We are in a world where agents are being used as a single word for the productivity enhancing agents that we have and also for the larger team and org scale workflows that we are building. And that is really confusing, right? Because if you're trying to understand how does this make a difference for me, the answer is going to kind of depend on whether you're really in charge of building those bigger pipelines at work.

It's like a an invoicing pipeline, right? Something like that. Whether you are in charge of enhancing your own productivity on your team. And so slashworkflows is not a command that automatically works for these larger pipelines, but it is a command that works if you're enhancing your effort as a claer, as a developer on your team. Your individual productivity can improve. One of the big questions of the summer of 2026 is how do those two pieces connect? If you are trying for example to build an agentic production pipeline for your engineering team, you have to ask yourself how do you allocate agents against a single source of truth whether that's your ticketing system or your repo in such a way that your individual engineers are productive but that individual productivity actually layers up to something larger. And that's where we see the larger unlocks is if you can start to think about how to write the entire rest of the pipeline post initial engineering work so that it is agent native first. That's when you start to get the unlocks as opposed to getting stuck in the sticky handoff. I saw this happen when I was looking at how Uber was complaining about their token spend and there's been a bunch of like highle folks who are leaking that they're upset about their clawed token spend. And the thing I called out is that there's a big difference between building agents for individual productivity and building an agentic pipeline that's native across the whole system and ruthlessly hunting for human handoffs that get in the way of unlocking that downstream productivity.

Because right now we have the piling problem. We have agents basically piling up a lot of work downstream and if you can't figure out how to manage that bottleneck as it moves through the system, you just have a giant pile for a human to review at some point in the system. There's there's no other way around it because agents are so good at generating stuff. And so when we talk about workflows, when we talk about 4.8, we need to understand that it is going to accelerate that if you don't have an agentic pipeline and and you need to think about it very seriously if you're in leadership, especially engineering leadership. How do you create an agentic pipeline that actually is more of a dark factory approach? A dark factory, just like a shorthand for you, is you submit the PRs from your engineering team and agents are the ones that handle the merge conflicts. Agents are the ones that handle the first, second, and third PR reviews. Agents are the ones that actually look through and monitor the results in production. Agents are the ones that review the the the other agents work in this whole system.

Everything is identified because if it is not, the work piles up unsustainably. And that doesn't mean that humans aren't involved. People think it does. It doesn't. It means that humans are increasingly over the loop. Not in the loop, over the loop, monitoring, designing the loop, and making it more effective. SLworkflows will just generate more downstream work for you if you don't think that way. And this brings me to the knowledge work piece. If you are a knowledge worker, if you are not a coder but like ending up being code adjacent because that's where we all are right now, then the best thing you can do is to think about your work kind of like that code stream and think about these new model releases and say, am I using these new model releases to generate downstream work for my colleagues in a way that's unsustainable or am I using them in a thoughtful way that enables me to accelerate toward overall outcomes in the business?

because that question is going to be the biggest question for the second half of 2026 for businesses. They want the outcomes. So, how do you get ahead of that and think that through? And when I look at that and I back that into the harness question that we've been talking about throughout this video, one of the things that's really compelling to me is that the codeex harness is more self-aware. And so, I can talk with Codeex and I can say, "Help me think through this outcome I want to drive with my team. I want to set up automations that enable me to do my work more effectively without generating unsustainable burdens. And Codex can strategize with you about that. Codex can think that through. Codex has the ability to do computer use and the ability to handle files in ways that help you think that through. I'll give you an example which requires one computer use in a way that Anthropic has struggled with. I know that the anthropic computer use score for 4.8 is very good. It's supposed to be better than 5.5, but in practice, codeex's harness with computer use actually works quickly and works dependably. And that makes all the difference in the world.

And I can ask it, use computer use, use your in inbuilt codeex browser and please set up this automation in such a way that the output format is not overwhelming to my colleagues who are doing ticket triage. And it will do that and it will give me the suggested template and we can align on it and we'll set up the automation. it will execute it reliably on my computer even when I'm away touching grass. Yes, I do touch grass sometimes and claude is just not there right now. And so my take for you on Opus 4.8 is that you need to be in a position with your AI work where you are thinking more about the harness than the model. And that's what this whole video has been about. It's been about the fact that 4.8 is very good and I will just say it again, very good at front-end design, very good at writing.

These are things that are classic strengths of the opus lineage and of claude more broadly. But you need to think about your goals, what you want to do, the outcomes you are driving and back that into whether that makes it a daily driver for you. And so if you are a knowledge worker, I would increasingly say ask yourself where are my outcomes coming from? If I am someone who needs writing help or who needs front-end design help, I would increasingly say if you're high volume, you're going to have to use codecs and write fat skills that cover those gaps or maybe work with Chad GPT like I did for the website. If you're not super high volume, then using Claude can make a ton of sense because it's just natively there and it's easier to work with. If you are an engineer, statistically speaking, and I've seen these numbers in surveys, 70% of you roughly are using cloud code and like 25 30% are using codeex and that number is shifting around. And then there's like this this like other section that's there that's a bunch of open source models. You should be looking at your tooling and making sure that your harness allows you to be productive in line with the overall outcome of the team, not just individually productive.

And that's one of the things I want to call out with SLworkflows is it's an incredible tool, but it's going to force you to think about that sooner than later if you're using cloud code, which like statistically speaking, twothirds of you are. And if you're using codeex, you still have to think about outcomes, but I love how self-aware the harness is there. If you are a leader, if you're a CTO, a CIO, and you're like, "Ah, Nate, come on. You keep talking about this Chad GPT thing. You were talking about Claude before. I'm so tired." I've got news for you. It is a two-horse race.

I'm not going to stop talking about the fact that both of them have a lot of strengths and that you should expect another you know lead in the race from for example Claude in in a couple months. You should expect your system to handle that. I will give you one more juicy detail here. These 10 trillion parameter models uh Mythos is in that class 5.5 is in that class. Uh there are others roughly speaking right some of them they're not admitting it but you can kind of tell if you use them a lot.

You should expect more open source 10 trillion parameter models by the end of the year. And so you should be architecting your system so you have the option to hit very strong open source models by the end of the year. It's just at that point knowledge work will largely be solved. So why not why would you assume you have to spend it on a particular model maker? Architect your system for flexibility. So that's my overall take on 4.8 very strong model.

It has a problem with consistency driven by overthinking and it is it is not fitting the harness as well as it should. And codeex in 5.5 is fit they fit handin glove from a harness perspective and I think that's really important to think about and I've tried to give you specific examples so you can actually see why harnesses matter so much. I don't want harness to be a foreign word. I want you to understand it's all of the it's all of the the scaffolding around the model so it can do its job and as the models get stronger you have to adjust your scaffolding to work and that's part of how codeex is strong right now is they've adjusted their scaffolding so we will stay tuned I'm excited for mythos if you want to get the full breakout of the tests that I ran for this head to the substack if you want a great great guide for which you should pick up for which thing because I've given you general principles here but like you should dive in and actually figure out which works for you, how to get started.

I have specific guides for you on Substack for both 4.8 and when to use max and when to use high and also for codecs in 5.5 so you can compare them and figure out what's best for you. And yes, you can obviously feed this to your model to figure that out as well. And that's one of the things that I put in the substack is like a guide that you can like feed the feed feed the thing in, have the conversation, figure out what's best for you in a conversation because so much of what we learn is conversation. Okay, I will catch you next time. 4.8 is one of the most interesting model releases I've run across and I I think it illustrates where we are in the race.