

The next paradigm shift (according to Karpathy)

20 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=T0C2N0B9LI0](https://www.youtube.com/watch?v=t0C2N0B9LI0)

<https://www.youtube.com/watch?v=t0C2N0B9LI0>

SUMMARY

Claude Tag is a new Slack integration from Anthropic that allows teams to interact with the AI model Claude as a collaborative team member, enhancing workflow and context management. This integration marks a significant shift in how AI can be utilized within organizational settings, moving from isolated interactions to a more integrated, team-oriented approach.

- Claude Tag enables Claude to join Slack channels as a team member, allowing users to delegate tasks directly within their conversations.*
- The AI builds contextual knowledge over time, adapting to the specific needs and workflows of different channels.*
- Claude operates in a multiplayer mode, allowing multiple users to interact with it simultaneously, fostering collaboration.*
- The system supports asynchronous task management, letting users focus on other priorities while Claude works on delegated tasks.*
- Claude can proactively provide updates and follow up on unresolved tasks, helping to manage communication chaos in Slack.*
- The integration is currently available only for team and enterprise plans, which may limit accessibility for some users.*
- Claude Tag's design emphasizes context management at the channel level, potentially simplifying the setup compared to traditional AI agent configurations.*
- There are concerns about reliance on a single model (Claude) and the need for flexibility in choosing different AI models for various tasks.*

This is a new paradigm for interacting with Claude that is significantly more in line with all the other human activity orwide. Once you do all of the under the hood engineering work to make this just work, Claude basically joins the team in a seamless way. You can talk to it as you would talk to another person and it can help with a very large variety of workloads. In my opinion, this is the third major redesign of LLM UI and UX. The first paradigm was that the LM is a website you go to. The second was that it's an app you download to your computer. The third one is that it is a self-contained persistent asynchronous entity with orgwide tools and context working alongside teams of humans. It really takes a while to wrap your head around it, but it works and it is awesome. Sounds like Karpathy's cooking something important here, right?

If I told you that the guy who helped pioneer LLMs as we know them today, one of the greatest researchers of our lifetime, one of the most important people in the entire AI world is talking about a slackbot here, you'd

probably think I'm insane or you'd think he's insane or you might think that he's drank the Kool-Aid at anthropic far too quickly as a recent hire and is just glazing a random at claude feature inside of Slack. And if you thought those things, I would understand entirely. But I have to do two of my least favorite things here. Actually, I have to do three of my least favorite things here. I have to defend anthropic.

I have to make a video about it. But most importantly, I have to talk about Slack. And I don't want to talk about Slack. Believe me. As cringe as this post may be, and it is, there are important things here that we can all learn from as an industry. And I think it's actually pretty cool. All that said, Claude Tag is only available on team and enterprise plans. So, if I'm ever going to be able to afford this, paying full token rates instead of my subsidized subscription pricing, we're going to need to have a lot more money.

And I'm going to cover a little bit of that real quick with today's sponsor. There's a bunch of things our tools do that we've just gotten used to because it's how it always worked. One of those things has always pissed me off. It's code review. Not the concept of code review. Believe it or not, I think we should be reading more of our code in general, even if agents are writing it. But the actual way that we do code review, it's terrible. Can someone please explain to me who thought it was a good idea to sort a giant code review like this with 11,000 lines changed in alphabetical order? It's nonsense.

Wouldn't it be much nicer if your PRs were broken up into logical layers that actually explained what each did and the files were actually sorted in a way that made sense based on the changes themselves? Oh, it's on the screen, isn't it? Yeah, Code Rabbit did it. They already know how to review your code really well. Believe me, I've had Code Rabbit review a lot of my code. This is probably the single company that's prevented the most production outages on my services of any single company in existence. Their stuff got so good that I found myself reviewing code less because actually reading in GitHub was so painful. And now they've solved that, too, by taking all of the knowledge they get from reviewing your code and using it to structure that same diff in a way that's actually readable. You can hover over any of these sections and see what they actually are. This isn't different commits. I want to be clear about that.

This is the whole poll request broken down into layers so you can actually read them and understand what's going on. And there's even little indicators saying which ones they left comments on. There's a great mini map on the side here so you can see which sections had comments and other things of note in it. It's sorted in a way that actually makes sense. Each section has a summary describing what it did which makes it so much easier to understand what's going on. And yeah, this is just really good.

It's time to start reading code again. Get started at sidyv.link/codabrabbit. So, what is this new Slack bot that Anthropic has put out and why is Karpathy so hyped on it? It's called Claude Tag. And the things that make it special are pretty damn cool if I'm being real. Claude Tag is a new way for teams to work with Claude. We're starting with Slack, which Claude can join as a team member. Grant Claude access to selected channels and connected to whichever tools, data, and even code bases you choose. Then anyone in the channel can tag Claude in and delegate tasks to it while they focus on other work. Claude builds context by remembering relevant information from the channels it's in and can plan out tasks to complete in the future. There's a couple pieces

here that are really important and know just that they're starting on Slack. As a Slack hater, I am excited for this to come to other things. The interesting pieces here are the way for teams to work piece and the channels it's in piece. These two parts are what make Claude tag so much more interesting than people seem to think.

This isn't just another way to tag an agent inside of your Slack. This is a different way of thinking about context management and tool access for real teams. And I'm speaking a little bit from experience here because my team has been playing with a lot of stuff like this for our Discord management of all of the different things we do. Whether it's the content on my channel, the sponsor deals that we're working with, the podcast and building topics and planning things out for that or the other creators that we're starting to help with their brand management stuff too. We have been trying to build bespoke Hermes agents and openclaw style agents that can answer the right questions with the right context in the right places and it's actually kind of annoying to get right. We're at the point now where we end up creating different isolates like actual containers that have all the things a given Hermes agent needs and then connect it to just one Slack channel.

But each of those agents is its own deployment that has its own everything that we built for it. Claude Tag is stumbling upon the same value prop here without all of that additional customization in a way that I actually think is really cool. What's even crazier is how much adoption there has been at Anthropic for this. According to them, tagging Claude is now one of the main ways we get things done at Anthropic. Today, 65% of our product team's code is created by our internal version of Claude tag. Very interesting.

So, how is this better than just using Claude code? I'll let them explain and then I'll give my thoughts on top after. At Claude is multiplayer. Within a given Slack channel, there's one Claude that interacts with everyone. This means that anyone can see what it's working on and can pick up the conversation from where the last person left off. This makes tagging Claude very different from working with a single chat or for a single task. It's much more like interacting collaboratively with a teammate. This actually is quite fun and it's amazing to me there are so few experiences like this already. We've kind of simulated this with things like PR review bots that can make changes where one person can at bugbot or at greile atcodeex atcloudcode at code rabbit or whatever and say hey can you make this change and they can propose changes and actually merge things into a PR. There are not that many experiences that allow that type of multiplayer that actually make sense for a fastmoving team in an environment that isn't [__] GitHub. If your multiplayer story is GitHub then you don't have a multiplayer story. you have a bunch of really slow load times in a website that crashes all the time. The idea of being able to talk with my team and Claude at the same time is actually really nice and I've started to feel this myself again with the cool agents that we've been spinning up for my team for the stuff that we're doing. This is something I actually really like about it. I'm excited to see how other systems and services start to develop these patterns in their own unique ways. The more important pieces are below though, specifically that Claude learns over time, not for the whole company, but for the specific channel. As Claude follows along with its channel, it builds more context around the work. This means that users don't need to explain things to it from scratch over and over, and Claude can eventually automatically learn from other Slack channels and data sources if it's given the right permissions. It does not report from private channels, though. This gives it the tacet knowledge necessary for it to provide the best possible work. This part's really undersold in my opinion.

Different teams and different channels need different context. This is a problem I've experienced myself and it's one of the things that's been nice about spinning up different agents on different computers that I'm running for different tasks. You can give different agents different context, but once you have cloud code on your machine, your options are global or project specific. That's a tough split even just for my code work. Sometimes I want to bring in these four skills in these two connectors. Sometimes I want to bring in none of that. Sometimes I want to bring in everything I have. Sometimes I want to add another 2,000 to 5,000 tokens of context to my agent MD for specific types of work. There is no good abstraction here. There is no cleancut way to split between people, projects, teams, orgs, code bases, and tasks where you can manage the tools and the context properly for a given agent for a given run. We have yet to even come close to figuring out what the right places to split and draw these boundaries are, but channels are a hell of a lot closer than any of the things I have seen so far. At the very least, they map more naturally to the way we think and the way we structure our teams. If Claude's memory is for a given channel, it doesn't matter what code bases the company works on or how the monorepo is split up or how the various different sub mini repos or whatever structure they have, microservices, whatever, it doesn't matter because the channel is where the context lives. So if one team works one way and a different team works a different way and they have different channels in Slack, they can work with Claude in those channels and it could feel entirely different. It would be possible for any two teams to have an entirely different experience here because the knowledge Claude has is various and very different in those two channels. There are other parts here that I'm a little less excited about, but I could see being useful, like the initiative piece here. If ambient behavior is enabled, I love that it's so stupid they even put it in quotes themselves. But if it's on, Claude will proactively keep you updated about whatever it thinks you might need to know. It'll flag relevant information from across the channels it's in and the tools it's connected to and follow up on threads or tasks that have gone quiet without being resolved. This could actually be useful considering how chaotic Slack is just like in general.

Claude is the member that'll keep you up to date on what's going on. That I could actually see. It also can work asynchronously. You can send Claude a task and focus on other priorities while it's working. It can also schedule tasks for itself, pursuing a project autonomously over hours or even days. We found this particularly helpful at Anthropic. We now spend much more of our time delegating tasks to many quads in parallel. You also can send it direct messages, too, which is pretty cool, but not my favorite workflow. I'm going to talk about this in a weird way. I'm going to do a much more in-depth video on this in the future. Let me know what questions you have about my setup so I can get it right. This is my Hermes agent. My Hermes agent runs in a Discord server dedicated just to it. I'm also trying to port it to Rust for fun. We'll talk about that another time as well. My Hermes agent is in Discord for a handful of reasons and not because Discord is my favorite app. Actually, I would like to do less in Discord. As you see with my history here, it is untenable at this point. But I'm using Discord because I like Discord threads so much. So, so much. I came around cuz I was not a big Discord thread fan initially, but I've grown to love them. Especially the ability to reply to one message in Discord, but also have a thread and the it's good. It's good for this in particular because when I set up OpenClaw, I didn't actually like it that much. I tried really hard and it was useful for a handful of things, but what I ended up doing with my OpenClaw was just set it up as a bot that could archive YouTube links and SoundCloud links when I sent it to them and put them on my NAS for me. And that's all I really did with it because whenever I try to do something else, the context would get weird and broken because it was just one thread. And that was my biggest issue by far with things like OpenClaw is the default setup would have you get one thread whether it's iMessage,

WhatsApp, Telegram, whatever.

You only had one running thread and that was the context being managed which meant that it would prune that context all the time. It would just not get things right. And if I wanted to be doing multiple different things at the same time, I would kind of have to massage the context myself. Like I have to start with remember how we did this two days ago. I want to do something similar for this instead because the context is everything I was doing in that thread. I had to help it pick and choose the right parts and what things to do and use. And when I added a capability to it, that capability was available for any task I asked about, even if the tasks or the things I wanted to do were unrelated. And I had a problem pretty often where I'd have like a scheduled task and I was in the middle of something else. So, I'd ask it like, "Hey, can you check my email for this thing?" It would check. I'm like, "Okay, does this email mention anything about that?" And it would just so happen to be 11:00 a.m. when I had a scheduled task run. The scheduled task would show up in the same thread and just break the context entirely. I personally found this like entirely unusable, and I ended up relegating my OpenClaw to like one task. Apparently, OpenClaw now also supports Discord threads, which is huge cuz I don't think any other methodology makes sense here for this. I love having threads for my tasks instead. Here's an example of one I set up that I'm going to start taking more advantage of soon.

This is a job that every day at 11:00 a.m. goes through the programmer humor subreddit, finds the five top posts that it thinks would be at all relevant to me and my audience, and then generates a page I can go to to see these top posts. And it's also the actual images, so I can quickly rightclick, copy image, and then go post them on Twitter if I want to. But this is a page that gets generated at 11:00 a.m. every day that gives me free memes to go post on Twitter if I decide to that lives in its own thread entirely separately from everything else I'm doing with it. And that's so nice. Where I want to go and where my team's already been going is the idea of breaking up different channels with different agents that have different capabilities. And that requires me to spin up a whole new Hermes agent with a whole new backing with all of those different pieces. But this is what it seems like Claude is getting right with Claude tag. The idea that it can create this itself. And that's also kind of what's happening with Hermes agent here. I didn't go into a terminal and set up a cron. I just told it I want it to do this thing with that 11 a.m. cron. I literally started by just saying let's scroll the top of it. Every day at 11 a.m. I want you to go through the programmer humor subreddit and find the top posts that would be worth me stealing and putting on Twitter. I tell it to do a test run and it did. Eventually, I got annoyed about all of the spam in the context in the thread and I just wanted a nice page I could open on my phone and save images from. So, I asked it, can you update this job to make the content HTML page using my HTML plan skill? The images should be embedded as image tags so I can easily save them on my phone. And then it made the change. And now I have these HTML pages. I can click that have the images that I can easily rightclick, copy, and go paste wherever I want to.

And this all exists without polluting any of the other stuff I am doing. And it's so nice. And the harsh reality is that I don't think most people will understand how to create a system like this and set it up themselves and especially not going as far as realizing they need different Hermes configurations or different open clock configurations for different channels they have for different purposes. like the one I have set up to manage my sponsor deals is very different from the one I have set up to help me plan what content I want to put out or to update my code bases for me or to go change what's going on on my codecs on the same machine. That's actually

one of the things that's been really nice with this is that my Hermes agent is on a computer that I also code on. So I can tell it to go make changes to my T3 code setup or to my codec setup. And it's actually been really nice having a Hermes as my like general do random [__] solution and then isolating codeex and T3 code to just be for code. But all of those boundaries and all of that config and separation has been my problem. And once you do it, you kind of get to see into the future. And that's why I'm excited by what Anthropic did here with claude tag. They are making it much easier to get most of the things that are cool that I did there without having to set it up and configure it and think in that separation yourself. The right primitives shouldn't require you to think about what available tools exist, what the context is, how the boundaries are set up. It should work the same way we work ideally. And in building this as a channel level primitive is actually really clever and I think will be the new norm going forward as these patterns become more popular. There is a problem though and it's not the tag part of claude tag, it's the clawed part. I don't want this to be just one model.

One of the really cool things with Hermes agent is that I can switch the model whenever I want. I played around with GLM52 with it and it did a pretty dang good job. I switched over to GPT55 and it did a really good job with that. I switched over to Claude models and had to pay cash for it because they won't let me do it through something like, I don't know, my \$200 a month sub that was sitting doing nothing. So, I moved over to doing that with direct paid usage inference. I even use it with Fable for a bit. It was really cool. But when I saw the bill and I wanted to go back to using my subsidized inference, I switched back over to the OpenAI models.

It's really nice to have setups like this that I can make suddenly feel more powerful by just switching the model. It was pretty crazy going from 54 to 55 and the model suddenly was able to do more and the same exact agent I had set up prior was way more capable than it was before. It was way faster and more accurate and got better task completion and it was just better and you could feel the difference. I don't think you should have to rely on one lab for that.

And I don't like the fact that right now it feels like your options are all in customization where you're setting up the Hermes agent in the Python environment yourself. Every channel needs its own [__] Docker image in order to have it be isolated properly and you're building up all of the skills and context and everything yourself and you can switch models or claude tag where a lot of that works properly by default but you have no control beyond what you can ask it to do and you can't really ask cla tag to go use a different model. Something I've been doing a ton recently is telling my agents to go use other agents. When I'm using codecs with GPT55, I know its API definitions aren't great and I know its UI stuff isn't either. So, I have taught my codeex, hey, when you're doing API like design and you're making an SDK that other things will consume or if you're building UI, call claude-p prompt and let Claude do that work or ask Claude to come in and give second opinions. You're not doing that with Claude Tag. So, while I am hyped that they are taking the cool UX that I've been experiencing with other things and making it much easier to access for real teams, I don't love this being an anthropic specific thing. And I'm very excited for other companies to build clones of this so that you don't get reliant on just one lab and the way they want to do things in the models they produce because you can get way better answers way more efficiently at much faster speeds if you take advantage of other labs and other models. I'll end on this framing from Karpathy which was a reply to somebody defending him saying that he shouldn't be getting clowned on for this. I think a number of people on the timeline didn't read past the title and made inferences and comparisons that are just wrong and then used it as an opportunity to take cheap shots. This isn't a feature

like some crappy Slackbot and it's certainly not a claw though it has some aspects of it. It's an org level harness. The difference will become clearer over time. I am really excited to talk more about this idea especially as we refine the org level agentic work that we are doing as a team not just on like T3 chat and T3 code but also for all of the management for all the other crazy stuff that my companies work on. I run three companies at this point so I'm seeing how this plays out in various different environments and anthropic is going in the right direction here. I can say that with 100% confidence. So Carpathy, I'm sorry you got clowned on so hard. kind of expected with the move to Anthropic right in the swing of them being as [__] as ever. But this feature is actually cool. Shout out to the team building it. Shout out to Lydia for the awesome launch video as well. I see where it's going. You are right. The future is roughly in this general direction. And to people who don't want to go pay exorbitant prices for a claw enterprise plan so they can test it out themselves, go put some time setting up OpenClaw or Hermes agent in your own Discord or Slack and you'll see a lot of the value that we're seeing here. Can't believe I just made a video defending both Anthropic and Slack, but here we are. Let me know if you think this is cool and how you're talking to agents with your teams. And until next time, peace nerds.