Building the GitHub for RL Environments: Prime Intellect's Will Brown & Johannes Hagemann

44 MIN · YOUTUBE · HTTPS://YOUTUBE.COM/WATCH?V=SJC1Y5Z5WWM
https://youtube.com/watch?v=SJc1y5z5wwM

If data is the bottleneck, if having the real expertise is the bottleneck, like would you rather have the smartest person in history work at your company or someone who's been there for 30 years? And sometimes you really want the person who's been there for 30 years. There's a lot of expertise that comes from really understanding a problem deeply and interacting with it over a long time. And this is really what happens in training that is almost impossible to replicate in a a short prompt. You really want the ability for institutional knowledge to compound over time, for best practices to compound over time. And this is how institutions and companies uh grow to be really powerful and successful is they stand on the shoulders of what they've done before rather than kind of resetting every day. Yeah. And we want to have this be accessible to any company that wants to do this. And I think that's how we thought about approaching it, especially as software becomes easier for people to manipulate, as the barrier to entry for [music] coding becomes easier, we see the same happening for AI research.

Will and Johannes work at Prime Intellect, which is one of the coolest new labs in AI right now. Uh your mission is to make frontier lab training accessible to everyone, which I think is a very noble mission. You have really really strong taste and just developer feel and just understanding how to, you know, that that intuition for what what developers care about. And then what you all launched with uh the reinforcement learning environments hub was like really really differentiated and people were very excited about. And so I'm excited to chat about many topics with you all today. Um post-training, reinforcement learning, agent harnesses, uh your platform, the RL hub, and then big picture questions on what's what's coming next in in post-training and RL.

Does that work? Sounds good. Absolutely. Maybe to get started, you are one of the leading research labs enabling customers to post train their agents. Can Can me about, you know, what what is that? What does that mean? Yeah, for sure. Happy to take that one and yeah, give a bit of a higher-level overview of like what our platform does as well as our research at Prime Intellect. Um yeah, so as already mentioned at the beginning, we try to make yeah, frontier infrastructure available to any like startup, enterprise, and yeah, new lab as well.

And uh yeah, basically the infrastructure that yeah, is currently locked behind the walls of the big labs where yeah, nobody really has access to them and um yeah, we really um start from like the compute layer and the compute orchestration layer and go all the way up then to the entire full uh post-training stack. So, everything from like the training frameworks that are needed to do large-scale reinforcement learning to um yeah, the environments with yeah, a bit of a more community approach with our environment hub to yeah, other pieces that are actually needed to do this like um yeah, sandboxes for secure code execution and um yeah, evaluations

as part of our environment hub as well. And yeah, to offer this like as an end-to-end um yeah, product in a sense.

And what's the intuition for for Why even why pursue that mission statements of making all that infrastructure available to everyone? Yeah, there's a lot of reasons. I think one is and I think something that we are very passionate about is just like open science as a way that humanity moves forward where like a lot of the big scientific discoveries historically have been things that we we talk about and as a world we can kind of build on top of but kind of more practically speaking as well, there's a lot of value in model customization where we are like the winning applications are using AI for a specific thing, for some agent, for some workflow where you also want to be able to deliver this at scale and with cost-effective performance. Uh and so really the way to kind of really optimize these systems end to end is to be able to have access to the model weights directly where you can then craft the model to be the best model for your problem rather than some model off the shelf where the crafting happens just in a prompt. And so, it's it's really just allowing a deeper layer of customization than what you can do at the prompt level.

And then is your vision vers- uh vision for the future then that you know, every company will be pre-training their own models, post-training their own models, uh fine-tuning their like what what do you think the future holds? We definitely think that every company will be an AI company, and we think most AI companies will want to have an AI research lab. And research can look like many different things. It can look like pre-training, especially if you're in a domain where you maybe don't just want text in, text out. If you want something more uh bespoke. Um in a lot of cases it will be like post-training agentic or otherwise uh kind of focused models for specific tasks and workflows.

Um and I think that's getting to the point where it's productionizable and cost-effective where you can actually make this very practical for people to do at scale for kind of the right shape of problems that people want to solve. Awesome. And then can you say at high level what your platform does? Yeah, so we have a full-stack research platform called Lab. And Lab is about giving everybody the ability to do the things that a frontier research lab can do internally, but for anyone in the world who wants to do this kind of research. And uh a big focus of Lab is the notion of an environment. And so I think a lot of people have heard the term environment in the context of reinforcement learning, and that's a big focus of it for us, which is that it an RL environment is encapsulating the things you need to do to do reinforcement learning, where you can have a model improve via trial and error. But it's also more general beyond just reinforcement learning. And I think if you haven't heard of an environment before, it's essentially the same thing as the evals that get reported when people talk about new model releases. So SWE-bench and AMIE and Terminal Bench, these are all examples of environments where there's a data set of tasks, there's a harness for the model to be in, and there's something called a rubric or reward function, which is responsible for grading the quality of the outputs. And so the same thing you'd use as an eval offline as your kind of your test set, uh you can use this in reinforcement learning as your training set. And this is a way to uh improve model performance interactively. And so our platform is really enabling people to use environments in their workflows for post training, for evaluation, for synthetic data, for reinforcement reinforcement learning. And it's also very much focused on as a community platform in the same way as things like GitHub are, where this stuff is new, it's complicated, and we want people to have lots of building blocks they can draw from and lots of examples and for it to be collaborative and for people to have reasons to kind of

show off ways of using uh models or different tools in workflows as environments to allow other people to post train models uh in those environments to kind of have this way of kind of sharing the ability to improve performance across different tasks.

Awesome. I think the general idea is to give more companies the actual ability and advantage that's currently only the big labs have in the sense of like this product model optimization loop where they um yeah, can optimize their models um for their specific product in a sense. And uh we see it as a yeah, thing where um yeah, that's the kind of reason why like a ChatGPT was created by OpenAI or like a Claude code was created by Anthropic. They actually have the capabilities to yeah, optimize models for their specific scaffolds in a sense.

And um yeah, where I have their models work way better in their products. And um the more popular those kind of products are to become in a sense, like a Claude code becoming extremely popular right now, um there the big labs have yeah, naturally less of uh an incentive to actually make it work better for like other coding startups in a sense, right? And um the idea there is to to give them the tools to have like their own like model product optimization loop. Yeah.

And um yeah, I think they're early adopters on that front. I think yeah, one great example um uh always in this case that I always give is is Cursor, for example, um that yeah, realized that in my opinion quite early on. They built their own like composer one model where they did like large-scale post training in a sense. And um yeah, really optimized a model where the environment was actually Cursor itself. So, uh yeah, gives them all the tools that like you have in Cursor in a sense as well. And um yeah, uh, optimize a model inside of Cursor. And yeah, we believe there's a lot of more startups that, yeah, will go in this direction to, um, yeah, on the one side optimize their current products in a sense, but also, um, yeah, uh, build completely new products that are really not possible right now without having this product model optimization loop. Awesome.

And can we say word on you said something interesting? Um, you said environments are just evals. Um, can we can we dissect that statement? In in my head an environment is a it's state. It's a description of world state. Um, through which, you know, you observe what actions you take, you observe how world state changes, and therefore you update your world model. That to me is distinct from an eval, which is like, you should have gotten this answer on this set of questions.

And so, can you help me merge those two realities? Sure, yeah. So, I think there's, um, a version of eval that's kind of where we were maybe a year or two ago where a lot of evals were like question and answer, uh, and it's like this big bank of questions. And then maybe there's other notions of environment that people think about when they talk about like, uh, I don't know, old school RL with like Atari that is much more about like this kind of long-running uh, state interaction loop. And I think where we are now is it's both in the same thing where especially the the environments that you want to do large scale training on, they do have this complex state. They maybe are simulating a web app. It's a full-fledged kind of coding platform where you have an agent doing these things, but in those in the original RL games, there's always a reward. There's a goal. Uh, and so, the the notion of there being a goal of this problem where it's not just running through some system and a human's going to kind of vibe check it. There actually is something that can measure progress and performance. And that

that's kind of what I mean by it's un-eval in that there is a system that it starts at some state, it uh, interacts with the system and the environment, the harness, the, uh, the agent, whatever you want to call it.

>> Yeah. Um, but there is some goal. And there's a way to measure whether it's doing well or not. Hm. Okay, got it. Um, and then is there a difference between you mentioned kind of Cursor as a great example of somebody who's doing really great frontier work uh in reinforcement learning. Um is there a good way to think about when should you be kind of constructing RL environments versus when should your actual application and your application states be the the environment, so to speak?

Yeah, I think there's definitely reasons where you want both. Especially like let's say you're training a model to be a really good Rust coding model. Here you might want it to be good for lots of different applications where you'd have different environments that are focused on some like domain task. Maybe you're a company that wants a model that's going to be really good at calling your tools or using your specific uh domain language that then you can provide as a service to people who are building around that. Um then there's also ones where the product very much is like a user interface for an end user where the user's interacting with an agent, in which case it might make the most sense for the product to very directly become the environment where I I think the the companies who might want to be doing this are the same who care about whether they're using Claude or GPT-5 or Gemini or the ability to choose models and have internal systems to evaluate whether a certain system prompt is good or whether changing out a model endpoint is good or whether using the mini version of a model is a better uh cost-performance trade-off. The infrastructure to do that is that a lot of people have already been building at these kind of advanced agent companies is the same infrastructure you use to do reinforcement learning. And so I think that's kind of this uh kind of convenient world we ended up in where the training paradigm that makes the most sense for improving model capabilities is the same sort of thing that a lot of people have been building up the muscle for just without doing the training piece. And so that's kind of where we see RL being a very useful tool for people to then have this as an option in their toolkit for system optimization. Got it. Okay. Um Let's talk about agent harnesses as they apply RL. I feel like harnesses like the the theme of the moment, especially with you mentioned Claude code getting so much love. I think one of the things that they do exceptional engineering around is the harness. Um harness and RL, are those things orthogonal, mutually exclusive? Like how do they relate?

They definitely relate. Um I think the way I think of a harness is like a piece of the environment and so there's some for any like eval or environment task there's some input a bunch of stuff is going to happen then there's some output state which is then going to be graded and so this whole intermediate piece whether it's interacting with some simulator or interacting with some other agent or physical world sim this is the environment and the harness is very much a piece of that couples how the model interacts with any other pieces of the system and I think depending on the application area like I think for coding agents we have a pretty clear like definition of what's the you could say the harness is the CLI coding agent and the the terminal is the environment but this isn't necessarily going to be universal across like all different types of agents in some cases it's a system prompt and some tools is the harness in some cases it's something that is going to be spawning sub agents and the sub agents also have their own harnesses and so there's a lot of complexity but I think the way we've thought about it is like harnesses are going to keep evolving there's going to be this Cambrian explosion of ways people want to use models and we want to take a a pretty general approach in defining what you could do with a harness and so

what we are really thinking about and why we use the term environment as the abstraction is like agent is too narrow harness is like kind of too narrow and you can do all of these things within an environment but the environment as a abstraction on the whole allows this any sort of system model interaction is in scope.

And then do you think then do you think all companies should be post training their models with environments are there specific kind of you know where the bullseye like you absolutely need to be using environments or is it like you could be post training with a different method versus you should just be prompting your thing? I mean I think environments are tools you can use to do all of these things and so I think that's one of when I talk about kind of environments beyond RL I think part of the reason why it's a useful abstraction is because it doesn't tie you to RL.

Let's say I want to have a small model and I want it to be distilled from a big model. The way you can do this is you take the big model, you plug it in your environment, you let it run a bunch of times. Now you have a lot of this data that comes from the same interaction protocol. You can use the same grader at the end to filter for the best examples and then do SFT fine-tuning on that. You could do prompt optimization with an environment. You could AB test different models with an environment you would with an eval. Um and so we really I think the idea is that every AI company should be optimizing their AI systems.

Mhm. >> I think that's a less like controversial take. Okay. And maybe there's another way to frame it. Like the the eval is almost how your agent performs in the set of environments that you expect your customers to face. Yeah, I think it um it depends on what you want to call like and then in kind of traditional machine learning terms, you don't want to overfit to the test set. Um and so in some ways we kind of are already accepting that we're going to be like using the test set or the eval to measure to have that kind of filter back into the model. Um and so it I think it's a little tricky to distinguish even like what is the eval, what is the environment. Um we kind of think of them as one in the same.

Um and we use the environment term very generically where an eval is a type of environment that's used for measuring performance but not training on. Um and that's kind of how we see a lot of people thinking about when they're doing evals, what they really mean they're doing is they have some way of measuring current performance and they're iterating on it. And in some ways RL is this iteration applied at scale where you're automatizing the process of changing the model a little bit, changing the prompts a little bit um and having this be the way that you can kind of hill climb on some goal. Let me ask you another question. Do you see reinforcement learning as synonymous uh from functionally with post-training?

Meaning like if your post-training model are you doing reinforcement learning or are you doing other things? There's definitely a lot of thing I think reinforcement learning is like the big thing now where it's like in many cases practically speaking, the if you're doing a large scale model, RL will be where you spend the most of your time and focus and compute. But it's not the only thing you want to do. There's a lot of things involved in the whole process of going from some initial model to the system you want to deploy.

This can be prompt tuning, this can be SFT, this can be online distillation. There's a number of algorithms that

all kind of are under this umbrella where RL is like the big one in the middle, but there's a lot of stuff around it. And uh really I think exposing this tool could be helpful to people and letting them have all these knobs they can play with is the way to unlock And are you finding that it's the, you know, really smart AI researchers that actually know how to make this stuff happen in practice? Is it, you know, towards your goal of democratizing AI development for all, you know, does your average Fortune 500 know how to use the platform and get value out of it?

I think most companies have people who can. Um I think any Fortune 500 any Fortune 500 company will likely have a team of AI engineers who are capable at following the like latest tools, who are good at using cloud code, who um have a lot of opinions about models and prompting. And those people certainly can do this. That's kind of the audience where we see as the target customer for this. >> Hm. Especially if you give them the right tools to actually do it. Um like maybe some of those large companies don't really have anybody in there who can like debug your TPU cluster in a sense to actually kick off such a run.

Or um yeah, other components that are needed in there to like actually just make it easier in a sense to yeah, do large scale like um agentic reinforcement learning with tool use, with code execution, and pieces like this. Um to yeah, just abstract away the the entire infrastructure for them. Yeah, got it. Um awesome. Actually on that note, do you guys have any favorite customer stories that you want to share? Um yeah, one of our favorite customers that I would like to uh yeah, point out in a sense is RCI, a new lab working on like frontier open models. Um I've been working with us on, yeah, the entire stack in a sense, right? Um, we've been talking a lot about reinforcement learning, right? Um, but yeah, we've been, yeah, also doing a lot uh, large scale pre-training in the past. Um, that's basically where our history is coming from in a sense as well, right?

So, uh, yeah, I've been training with them some of the largest, uh, yeah, mixture of expert models, um, and actually able to, yeah, open source them as well and, um, yeah, as well on the on the post-training side, um, with them as well. Um, maybe, uh, Will, do you want to share some more? Sure, yeah. So, they're a very close collaborator of ours that we've, I think, uh, we've all been friends for a long time, but also like we've been, uh, I think they've been way where we've had the they've had a lot of things that they want infra for and it's that's been a way that we've been able to kind of, uh, force us to build out a lot of the pieces both from computer orchestration to, uh, post-training to pre-training to inference, um, around kind of just making the everything that is needed to kind of be a Frontier lab.

Um, and I think they're very aligned with us in the kind of the openness mission, um, and I think but I think they're kind of focus they are more targeted at like enterprises and the end user where they are kind of going to work more directly, um, with customers in terms of like end-to-end, uh, kind of, uh, delivery of a certain artifact where I think where we come in is we are, uh, we, uh, are really focused on the developer experience at the infra layer and the ways that we can make put these tools into more people's hands where the process of going from idea to deployable model can become as, uh, seamless and kind of quick as possible and efficient as possible. Awesome. And then any other favorite customer stories maybe on people that are using the Environments product specifically? Yeah, so, we are definitely really focused on like, uh, the research community and like some of this is like, a lot of grad students use it, a lot of like students and people who are getting, uh, early, uh, in

like their career learning how to do this stuff are using it, but also a lot of labs who are focused on a very specific domain where, let's say you're starting a medical AI lab. Uh, we work closely with a number of groups in the medical AI space where they want to create more uh both uh benchmarks to understand how good our models are at medical capabilities both in terms of diagnosis or patient interactions or question answer about medical literature um or agentic search over certain medical tasks. Um and so like so far OpenMed being two that we work closely with um where the focus there is to they really care about like earning trust from the medical professionals. Uh and so for them they really want to focus on the domain and not as much on kind of the the generic medical uh the the LM infra that is kind of like a headache for a lot of people.

And so for them like being able to kind of have this platform for creating evaluations, for showing them off, uh for being able to use them to then improve model capabilities that could then be deployed locally in a hospital or uh deployed locally for some end user where the ability to have this customization and have this kind of end-to-end trust and understanding of the data provenance for tasks at hand or the ability to kind of customize models very directly is very key.

Do you have any customers that are using you for more of like what you call the old-school kind of Atari style, you know, learning from your environment type? By all do you mean like so I when I think of Atari I think of like non-LLMs. Um and so we definitely are focused on LLMs and uh foundation models that look like LLMs. Uh there's definitely a lot of researchers who uh use our platform for uh these things that are more like there's some examples we have on the platform that are much more like games.

So I mean kind of one fun one that I use like a demo a lot of is the game Wordle from the New York Times where this kind of ended up just being the like a great hello world environment for people because it's the infra is really simple but it's very expressive where you can kind of get a feel for it and you get this aha moment of seeing a model learn to think about the game as you give it rewards for doing better and you can have you can do it with a really tiny model too. Like it's it's in the sweet spot of difficulty where you can do these runs on like a couple GPUs in an hour uh and see a model actually learn how to get better at the game and Yeah, I would say like the more toyish game examples yeah usually the ones that people are going for for actually learning how to build those reinforcement learning environments and yeah that's also what people are heavily using the environment up for in a sense just because we have all this infrastructure built around it to yeah be able to actually test and your environments and um that's usually how they start out and then yeah go to actually building more complex environments uh later on. Um another group I would love to give a shout out to in a sense that have been building some of the more complex environments on the hub people part of our reinforcement learning residency um where yeah we have like a group we initially started out with like eight to 10 people I think 14 to 16 are now in the group you have people grad students as well as yeah people working full-time that part-time like building reinforcement learning environments as well as doing novel research on top of the environment hub and yeah those folks have yeah built yeah amazing yeah environments in like all kinds of verticals in a sense everything from like verifiable software engineering lean to medical physics environment to some cyber security environments so uh and then yeah also we give them the tools obviously now to to actually do the training in those as well. Yeah awesome.

Um can you help demystify for me I've heard that some of the foundational model companies are spending you know millions of dollars each on some of these environments. >> Yeah. Um and so like you mentioned cyber security at the end so like what goes into constructing a cyber security environment? Yeah so there's there's someone in our residency program who actually does a lot of the stuff and so like there actually is a lot of like tooling in the cyber security world around like these kind of capture the flag games where there's there's some system that has some hidden bug in it and this is like a challenge that pro originally these are built for programmers or programmers will have like a little hackathon where they try to go find the bug in some system but you can adopt these challenges to LLMs as well where it then is a a full software environment, where it's a terminal where the agent lives in the terminal and has access to tools for doing bash commands. Maybe it's using Claude code or some other wrapper for an agent harness, but it's in a terminal full of files and can interact with these files. And then at some point the it marks that it's finished with the rollout. And then you can grade the state of the environment using other pieces of software, other code that executes.

Um, but we do like we actually have a lot of people we work with who are in the data and environment space, where we found that um, there's a lot of interest in using reinforcement learning as a way to evaluate data quality, where um, the ability to measure what happens when you train a model on a set of tasks, set of rubrics, set of environments, uh, allows you to understand bugs in the environments because uh, there are issues that come up in reinforcement learning where maybe if your environment has a a backdoor, a model can exploit this and kind of game the game the system. And so I think there's a lot of interest in people using RL to like in the pipeline from like idea to environment that ends up in some frontier labs, uh, like foundational training run for the next GPT or Claude model. There's a lot of vetting that goes into this. And doing these like smaller or like medium-scale runs in like let's say one environment allows you to really poke and see where the problems might be. Okay, super interesting. And then we take the cybersecurity environment example one step further. By the way, I have no no particular affection for cybersecurity, but it's something that like I I love having a specific example in my head.

Um, I could see how you could construct a toy example, right? These are capture the flag examples. I would imagine they're toy examples. I would imagine they look nothing like the actual kind of corporate network environments of a, you know, real company with all of its cybersecurity products. And so I guess how do you construct an actual, you know, does does what people can construct, does it actually scale up towards imitating and reflecting the complexities of the real world? Almost like the, you know, the robotics sim-to-real gap?

>> Yeah. Um, is there a sim-to-real gap in crafting these environments? And is solution just to like train on like real kind of corporate security environments? Yeah, so I would say there's less of a barrier in terms of the actual complexity. Like the in principle these can be as complex as you want. Um it's anything that could be on a computer. So just think of anything that's on a computer or network of computers as potentially an environment. What really kind of uh becomes the bottleneck in many ways is like cost of the simulator where I think there's a lot of focus on identifying clever ways to kind of mock the right piece of the system where let's say you have an agent that you want to like let's say the internet.

Like let's say you want to have an agent that like does a lot of like web search. Yeah. In some cases you actually

want to do the real web search. In some cases you want to find ways where you can design tasks where you actually don't need the full thing. Um it's like this kind of uh I don't know um I think of like the map games where there's this world you want to explore where there might be this whole map and like some of it's like dark but as you walk around like the light shines then you now see this part of it. And so if you know which pieces of this map might actually be explored on a given task then this kind of allows you to decide which pieces are important or not. And so like one example uh like there's some bench there's a benchmark called Tattoo Bench that is very popular in the the eval world where it's about customer service agents involving a database. And so the database is something where like in a real system you might need the full database but if you know that the agent is only going to ask certain types of questions for a task uh you don't need a full database with millions of records or thousands or or billions of of uh like different examples. You can have kind of a mock database that is uh a cheaper to run maybe in memory piece of software that um only has the things that are expected by the agent or expected in the the scope for a given task. And so identifying like the pieces of the system that where the complexity is kind of overkill uh is part of this task design process to enable efficiency. And does your platform help with that? Like it it almost seems like you know, if creating environments is the bottleneck to training these systems, then like being able to efficiently create these environments where, you know, you're lighting up the part of the map that's we lit lit up is like a core kind of almost platform competency. Do you guys assist with that?

>> Yeah, I mean, it's definitely a way that we think about the design of everything and we kind of go down a lot of these different uh like rabbit holes as we have to focus on different tasks from working with different people. And so like coding agents maybe is one example where like there's a lot of complexity that comes up when working with um like sandboxes and terminal states and ensuring that you have like good snapshotting and all these things and protocols to interact with different agent harnesses. And so we've built a lot around that, but it's also the sort of thing that we kind of know that the space of complexity is going to grow arbitrarily over time as people start uh getting more uh deep into these different domains. And I think we've tried to design everything in a way where we we keep a lot of doors open where like we have room to build features that's kind of like there's there's base layers of like generic environments, then you can go I want a coding agent environment. I want a coding agent environment with a sandbox that's global across the run. I want one per rollout or there's lots of these different branches you can go down. And so we kind of have anticipated like there will be a lot of these branches.

There's some that we've built for. There's some that I think we're kind of ready to build for when we need to. Um and we also like think a lot about how do you make a good developer experience where let's say think about just documentation or skill files for agents. People are going to be using coding agents when they're building these. And so uh there's a lot of institutional knowledge that gets built up when you're doing this kind of research both for like a specific project or as a research team doing large-scale training runs.

And being able to surface this information, some of it is directly in the product. Some of it is in the way we design the libraries. Some of it is in documentation or skill files that get shown to agents. And this is the sort of thing where we we've designed it to be something that can evolve over time as like the research literature evolves as the uh best practices for different types of complex agents become more clear. I assume you guys read Sutton's um uh learning from What was it? Age of experience?

>> Age of experience, yeah. Scale AI era data labeling. Um do you think that constructing environments is sort of the natural successor to that? Yeah, I mean it it very much seems like it kind of already is, where it it does seem like a lot of the focus from the major labs has shifted to They're still using a lot of human data, and so human data doesn't seem to be going away because creating these environments kind of by definition is for things that models aren't good enough at yet, which means the humans are are better than the models in some capacity. And so identifying which pieces of information the human can most uniquely assist the model in improving its skill on, I think is really the key to target, which is how do you create this information flow from the human who really has the expert knowledge on how to do some sort of thing, what does a job well done look like, and get it into the model. And it does seem to be that RL is the most effective way to do this right now, where having tasks with a prompts to grade with an LM judge, a rubric for what success looks like on a task, is kind of the paradigm that's emerging for a lot of these cases, where a human Sometimes it's through a reward model, but kind of the most direct visceral version of it is There's a set of questions about yes no, was this done in the LM's answer, and that turns into the reward score. And so that requires a lot of human data. Okay.

Awesome. Um why create a hub for environments? Yeah, I think the hub idea um started by seeing a lot of different like open source repos out there that had like overflowing implementations of all those environments in a sense, and um yeah, and in general like we already or we'll already created the nice uh verifiers framework um before we even started the environment hub, right? Uh where you had different inside examples for environments in there, and it was very much a um yeah, an approach to like standardize the whole process even more process even more. Yeah. Um and beyond that, um, just uh, beyond just like sharing those environments, right? And having like an open source platform for it, it's also about having like a place where you can build a lot of, um, yeah, infrastructure around them, which you can't really do if you just upload them to like a GitHub repo or something like this. So, um, yeah, having a proper evaluations, uh, like integrated with your environments, so you can like immediately, um, test them across all frontier models is one of the features, um, yeah, people are heavily using, um, the environment hub then for, right? Um, you make it extremely easy to install one of those environments inside of, uh, yeah, a uh, a different trainer. So, um, there we obviously have our own like large-scale trainer with with Primer RL, which we've been, yeah, heavily optimizing for this, but yeah, we are very open there on the open source side to integrate with like a bunch of different trainers because people have different, um, yeah, needs on on the trainer side as well. Yeah. And, um, yeah, that's how how the whole idea of the environment hub, uh, yeah, um, And what is the community behavior then?

Like, do you see people forking, modifying these environments? Do you see them, you know, putting something 80/20 out in the public domain and then like, you know, we're going to keep our secrets for ourselves and, you know, not share that back. Like, what's the community behavior? Yeah, I mean, there's definitely a lot of people we work with who want to keep their environments private, as you understand. But the value for them of it being a hub is that they can do ablations on ones that are kind of known to be they can compare their private one versus some public one that might be on a similar type of task.

Or for evals, there's a lot of value in having kind of, uh, mixed in uh, uniform implementations of popular benchmarks in a way that if you're doing a training run, you can plug in some known eval as a way to monitor the progress of your run. And so, maybe you're doing well on your environment, you can see does your environment also generalize to other tasks. And so, having all these other tasks available is a really helpful way

for people to be able to, um, understand not just their own task, but other things they might want the model to be good at as well. Yeah, awesome. What are the most popular environments on the hub today? Yeah, I think it tends to be the ones that are these kind of uh, one is the ones that we use as the examples in the documentation that naturally in software tends to be the most popular ones.

>> Wordle stuff? The Wordle is one popular but I think one that we see a lot of like interest around and people one where I think there has been the most degree of people kind of branching and forking and turning it in different versions is one that we call Wiki Search which is doing search over Wikipedia pages but it's designed to be this template you could use for agentic search more broadly. So the there's a lot of applications where people want agents that know how to search over their internal documents or documents for a specific type of information and having this kind of template for if you just you just have to swap out the documents. And now you have this environment ready to go where the the rest of it is already kind of set up that tends to be the sort of thing where we see a lot of value in people being able to bring other types of documents that they'd want to do agentic search over.

Yeah, got it. Okay, awesome. Um, I want to maybe shift towards future research, you know, big blue sky questions. Maybe the first one Andre I think is one of your one of your angels as well. He kind of has that infamous quote of you know, RL is you know, it's it's amazing but it's it's quite inefficient and it's like sucking bits from a straw. I guess do you agree with that and what do you think is going to happen in the research side to make RL more efficient?

Yeah, I mean I think um it's definitely true that RL is using a lot of compute to get a pretty kind of small signal in terms of pure information but I think in some ways that's part of the value of it as well and I think one of the reasons that a lot of the the labs have really focused on it is that one of the bottlenecks that's hard to scale is human data especially high quality human data um and RL allows you to kind of trade off compute for data in a sense where you can get a lot of value out of a smaller amount of data by using more compute um and so the supervision comes coming from this data is like small that you can get a lot out of it more so than you can via pre-training or supervised fine-tuning alone. Uh as well as um it's useful in cases where you don't necessarily have golden examples. So, like if you have a bigger model to distill from, that's great. But, if you're already at the biggest model size that you have access to, um then you kind of need to go into untrodden territory, and exploration is really the heart of RL. It's how do you explore and try out different things?

And maybe there are ways to do exploration that are more efficient than RL that people will kind of discover, but this is kind of currently the the frontier of using compute to explore and improve capabilities. And so, it's what we got for now. For sure. And I think uh like I can't speak for Andre, obviously. Um But yeah, I would be curious to hear his views like 2 months later after the latest like Dwarkesh podcast in a sense on um his views, especially on the coding side for uh like uh large-scale reinforcement learning and how it actually helps there. Um if we look at something like like Claude code, which definitely was already popular before, but definitely popped up more uh over the last uh like 1 month, I would say. Um then yeah, my yeah, his views might change in a sense on like uh the specific piece of how uh yeah, useful reinforcement learning can be in the in the coding domain. Yep.

>> Um but yeah, generally we we don't think that's going to be the end in a sense, right? Um we generally think we want to be always at the frontier of like what comes next in terms of paradigms. Um yeah, there's definitely lots of low-hanging fruit still on the like pushing agentic eye capabilities even further. Um but uh yeah, we also like uh know there's limitations in a sense, right? Like some of the pieces we've been working on um where we definitely see limitations is on the yeah, context side. Um so, um yeah, I think there are um yeah, we we just like have a hard limit right now of how much uh like uh tokens we can fit into a context, and you have been thinking there about ways on how to actually improve that in a sense. Yeah.

Awesome. Um switching gears a little bit. Open source, open weight models. >> Mhm. Um what do you see what do you see the role open open-weight models play? Does your infrastructure um kind of work only on or work optimally on open-weight models? Could you kind of help people do post training around the closed-weight models? How does that all work? Yeah, so in in many ways like the the trainer itself is going to require having access to the weights and so if anyone who has closed-weight models wants to use it, we have we're happy to chat. Um but more broadly the idea of the environment is general across different types of optimization and so we can do use the the same infrastructure at the environment level for doing evals on uh closed models, for doing prompt tuning on closed models, for doing model selection, for evaluating agent harnesses. There's a lot of research that can be done around closed models just by having a way to do this experimentation.

>> Yep. Uh and so it whether using open or closed models, um it's or you can create data that some platforms might let you upload some examples that maybe you're distilling from a particular model into another um where you use the environment as this data engine. Um so there's a lot of different ways you can use the tools to optimize models. And do you need need need the weights? Like for example, can you kind of Laura uh Yeah, so I mean the Laura I would consider still part of the fine-tuning process and that's what we recommend a lot of people do for RL anyways. Um and so like you don't need that necessarily with the full weights but like you can't I can't upload my own Laura for GPT-5.

Um but I could bring an environment to the platform potentially. Um and so I think there's a lot of different ways you can do partial customization and I would imagine that in many cases the degree of like how many do you need Laura adapter do you need full fine-tuning is going to depend on the training recipe as well as the goal of your optimization. >> Yeah. What about can you do reinforcement learning on like the agent harness around a closed model?

Yeah, so I would definitely consider this in the domain of like like there's a a few world of prompt optimization that some people have uh been exploring in the research world that is in some ways kind of an analog of RL but in prompt space where Like the DSPY >> Yes, exactly. And so like the GEPA algorithm got a lot of attention like like last year as kind of a a what seemed to be a better way of doing this. And so we support we have support for that as well with our environments.

Yeah. Um where you can do this around different pieces of the harness where this might be what's the prompt used for a certain tool, what's the agent skill, what's the system prompt. There's a lot of these things that you can

apply different types of optimization to. Awesome. While we're on the topic of DSPY, um I think the DSPY authors also have this new thing that is the current thing. Uh what was it? Recursive language models? Yes. What do you guys think?

Yeah. Um yeah, definitely very interesting that um yes, I um yeah, already said earlier in the sense we are very interested in like um yeah, longer horizon agents and so on and like actually solving things for those um to move a few use cases and uh yeah, been internally thinking for a long time about uh like how can we um yeah, have models learn how to manage their own context. Um so um right now people are building a lot of scaffolds for context management and yeah, we believe uh like something that is a bit more bit bit a little bit in the sense is to have the model learn how to manage its own context. And um yeah, I've been yeah, searching for different research in this kind of domain for a pretty long time and yeah, the recursive language model uh research direction is one of the yeah, most promising ones in our opinion. Um we've been yeah, uh since yeah, Alex saying who's the original author of the RLM work um yeah, published it. We've been yeah, very interested in this kind of work, have been yeah, exploring it um as part of our like research as well. And um yeah, there were blog posts out um couple of weeks ago that um yeah, basically showed um using this RLM harness where you um yeah, pretty much give a language model access to um a variable in a persistent Python wrapper so it can not have this whole context or the whole data as input in a sense but it has it in this variable that it can then yeah retrieve it can transform it and yeah manage its own context through that and then also call other like sub elements where the recursive part comes from to actually like manage it and yeah that's the whole idea behind the recursive language model we've been doing yeah some exploration there on this front to actually just give current like frontier language model access to this yeah specific LM harness. So not necessarily training in this harness but just giving it access to this specific way of like dealing with its own context and yeah it's been all be shown to like improve benchmarks on yeah very long horizon reasoning quite a bit and yeah we are very excited as like a new frontier in a sense to actually train in this as well to let the model train well train the model to actually use this harness and yeah that's what we're going to work on over the next couple of months. Super exciting.

What else in the research domain? You guys have great research taste. What What do you think is on the horizon? I'm really excited about synthetic data research and it feels like there's a lot of stuff that feels like we should be able to do it but you haven't really seen it emerge in the open in terms of creative ways of doing kind of self-reflection um and I think people talk a lot about continual learning as this kind of idea that we're we're going to have to get better at models kind of learning things on their own and I think the idea of um using uh other tricks that we already know in conjunction in different ways things like prompt optimization things like uh distillation um in conjunction with synthetic data it feels like there's a lot of I don't want to kind of go too deep into different directions, but it seems like there's a lot of room for exploration around having models curate their own training data, maybe curate their own environments, um, and understanding which versions of this are most effective for like lifelong learning.

Love it. Okay, we're going to close on an optimistic note. Um, if everything goes right, what does the world look like and what is the role that Prime and Select serves in that world? Yeah, good question. Um, how would I answer that on a on a high-level overview in a sense? Um, I would say, uh, we don't want to have a world where like all the like future value of AI in all kinds of verticals is just owned by the big labs. Um, we, uh, have

something where we like empower like entrepreneurs and enterprises and so on to actually, uh, not get steamrolled in a sense and, uh, like optimize their products, uh, um, yeah, better than they have the tools, uh, for doing so right now. And, um, yeah, just enabling this and, yeah, a lot more, uh, like Claude Claude moments, a lot more Cursor for X type moments, um, that will be enabled through this.

Every company is an AI lab? Kind of. In some ways, yeah. I mean, if if data is the bottleneck, if having the real expertise is the bottleneck, like, would you rather have the smartest person in history work at your company or someone who's been there for 30 years? And in some ways, sometimes you really want the person who's been there for 30 years. There's a lot of expertise that comes from really understanding a problem deeply and interacting with it over a long time. And this is really what happens in training that is almost impossible to replicate in a a short prompt, where, um, you really want the ability for institutional knowledge to compound over time, for best practices to compound over time. And this is how institutions and companies, uh, grow to be really powerful and successful is they stand on the shoulders of what they've done before, rather than kind of resetting every day. Yeah. And we want to have this be accessible to any company that wants to do this. And I think that's how we've thought about approaching it, especially as software becomes easier for people to manipulate as the barrier to entry for coding becomes easier. We see the same happening for AI research.

It's really inspiring vision for the world. Thank you guys so much for joining today. You've really paved the way on environments and your environment hub and thank you for taking the time to demystify [music] what an environment is and and share your vision for the future. Thanks. Thank you.