Scholars & Big Models:
How Can Academics Adapt?

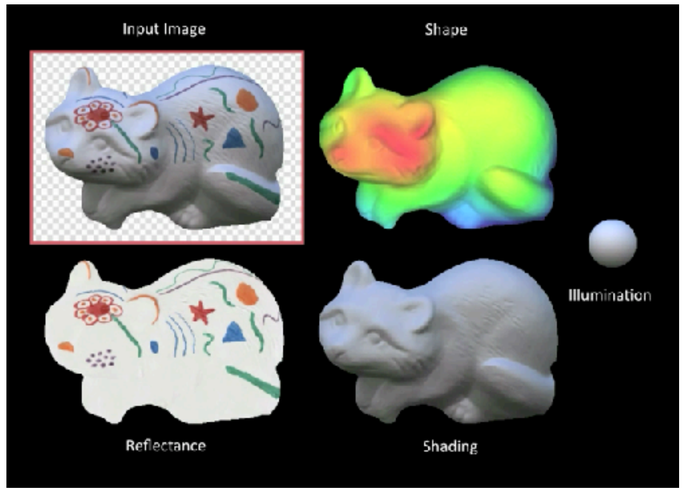
Jon Barron
Google Research

Hello! This is a talk that I gave at a CVPR 2023 workshop on “Scholars & Big Models” at CVPR 2023 (https://sites.google.com/corp/view/academic-cv/). The talk wasn’t recorded, and I didn’t have a script, so I’ve tried to reproduce roughly what I said during the actual talk at the bottom of each slide.

Curriculum Vitae

UC Berkeley
PhD Student
2008-2013

Google[x], then Google Research
Research Scientist
2013-Now



Just to quickly introduce myself: I did my PhD at Berkeley with Jitendra and I focused on low-level vision problems, like inverse rendering with end-to-end gradient descent. This was a tricky thing to do at the time, as we didn't have autodiff yet and everything was on CPUs and in Matlab, but we had some nice results. After Berkeley I went to Google and focused on other low-level vision problems like color constancy and depth estimation, and eventually circled back to that same sort of "inverse rendering with end-to-end gradient descent" problem with NeRF, which is all I do now.

It seems like I'm the only speaker today who doesn't have an academic affiliation, so I suppose I'll be playing the part of "the industry guy" today.

Obligatory LLM / AGI Takes:

- “AGI” and “Superintelligence” are non-scientific concepts.
- Intelligence is not equivalent to reading and writing words on the internet.
- I am just here to make 3D models and pretty pictures, I do not care what is or is not a cat.

My understanding is that we’re supposed to use this workshop to drop all of our hot takes about LLMs and AI, so let’s get into it. First off, I am not a fan of “AGI” or “superintelligence” as concepts — they don’t seem to be falsifiable, so I don’t really think about them much when I’m wearing my scientist hat (though for what it’s worth, “AI” and “intelligence” are similarly non-scientific concepts that I also don’t care for).

I think that it’s a little sad that the rise of LLMs has caused people to view “intelligence” as being exactly equivalent to “the words we read and write on twitter and reddit”. To rephrase Rodney Brooks: “Elephants don’t play twitter”. But others here are better equipped to comment on embodied intelligence and robotics than I am.

Fundamentally, I’m just not really much of an AI guy. I got into this field because I wanted to make 3D models and cool pictures, and I’m just not very interested in the underlying semantics behind an image. I want to know how far away stuff is and what color things are, and I’ve never really thought that “semantics” or “AI” are the right tools for those problems.

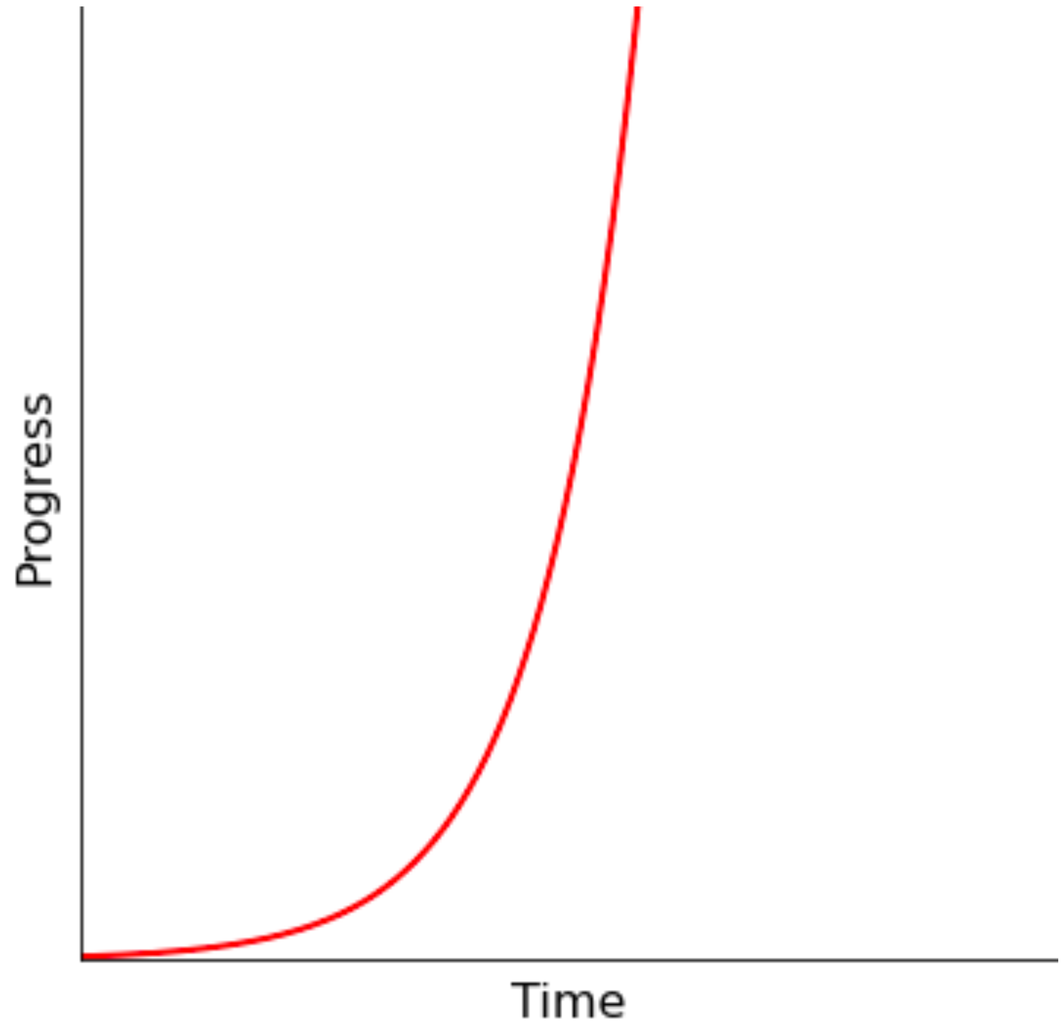
Obligatory LLM / AGI Takes:

- “AGI” and “Superintelligence” are non-scientific concepts.
- Intelligence is not equivalent to reading and writing words on the internet.
- I am just here to make 3D models and pretty pictures, I do not care what is or is not a cat.
- LLMs are a super important technology, and this is the singularity.
- Even if you don’t care about AI and LLMs, LLMs are very strong evidence for scaling maximalism in AI.
- We can and probably will solve vision and robotics with a similar approach.
- It’s okay if this bums you out.

All that said, I feel comfortable asserting that LLMs are a hugely important technology, at roughly the same level as the internet or mobile computing. I also think that we are currently in “the singularity” (in as much as such a thing will ever exist) — we have a clear line of sight between where we are now and where we need to be to achieve most of the non-robotics-related goals of AI, and this wasn’t true 5 years ago.

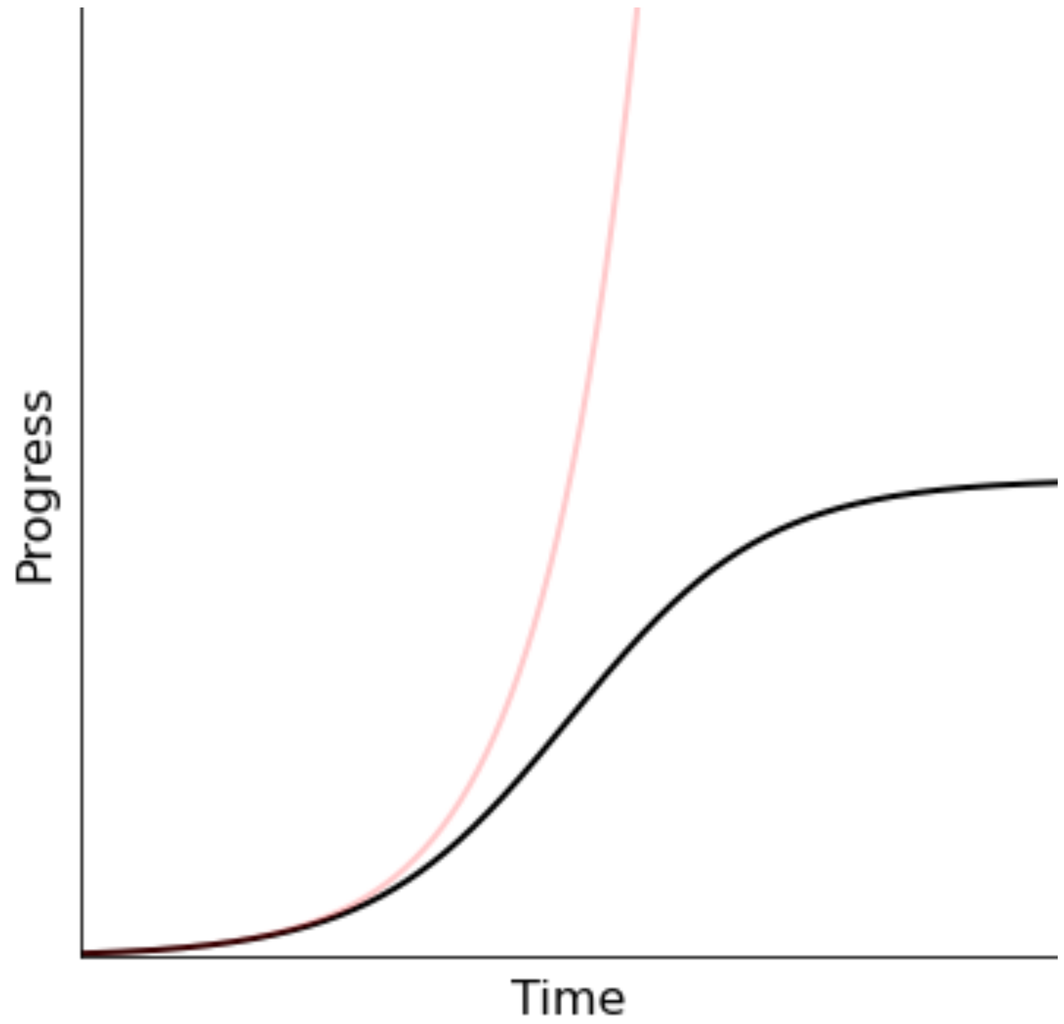
I also think that even if you don’t care about AI, you’ve got to accept that LLMs are extremely strong evidence for the idea that scale is a viable way to solve AI, and I think we will see similar solutions to vision and robotics using this general approach in the coming years. It’s okay if this makes you a little sad, a lot of the people here don’t want this paradigm to win out, and I get that.

How Technology Works?



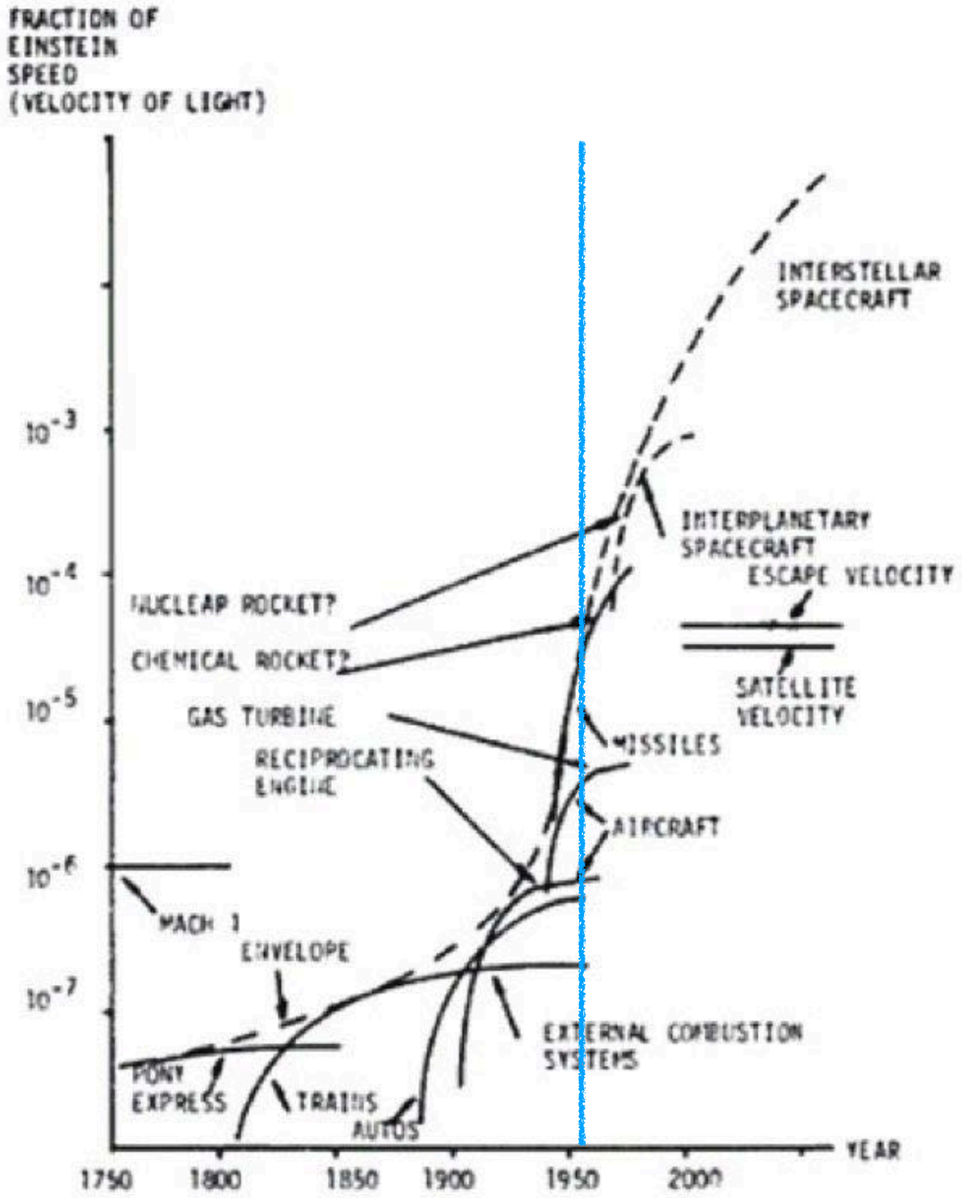
When people talk about AI and technological progress, they usually have a plot like this, with time on the x-axis and progress on the y-axis, and with unending exponential growth. I always thought this was pretty weird, because technological progress has never looked like this in the past.

How Technology Works.



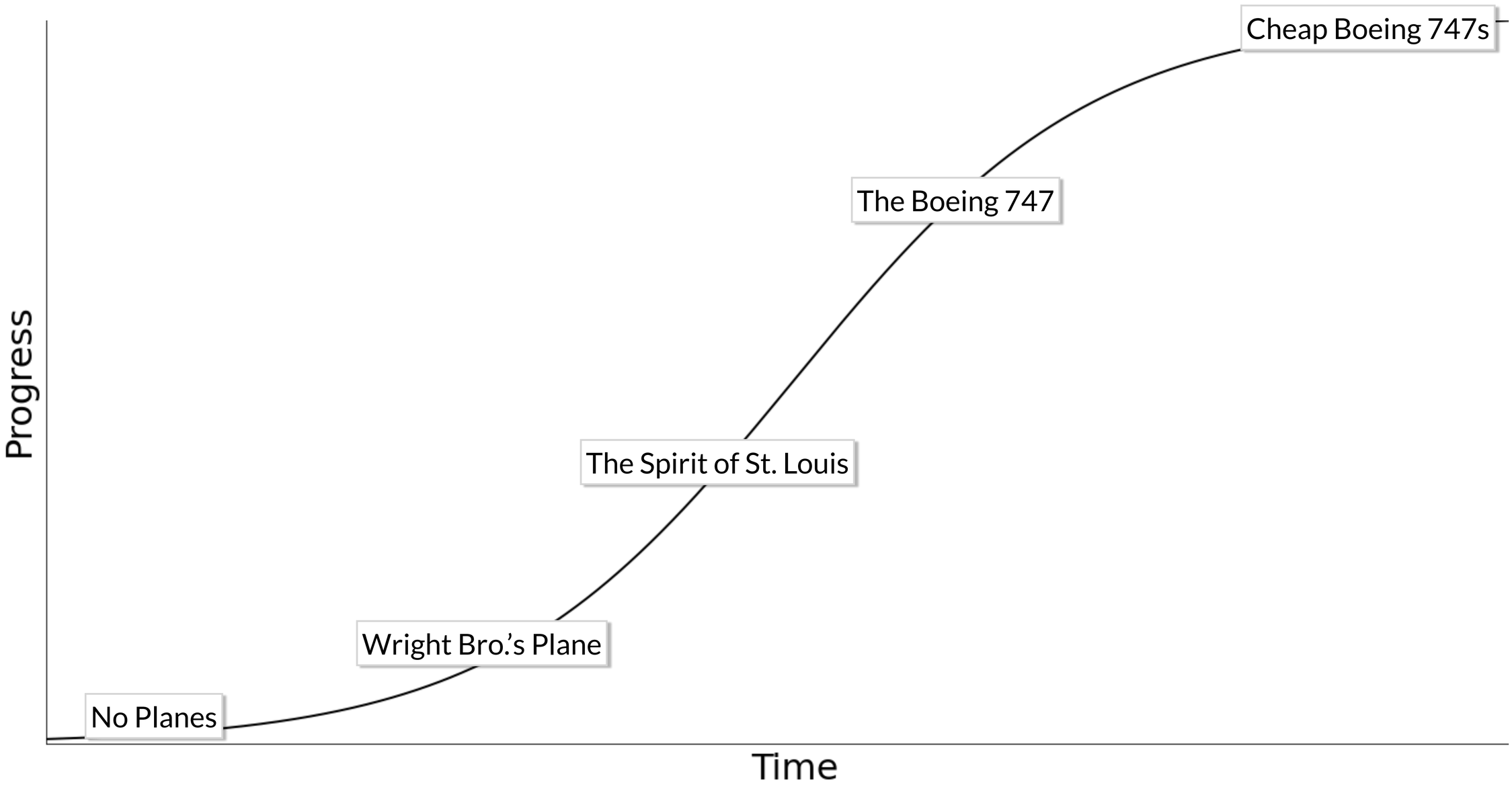
The progress of any individual technology up until now has always looked more like this: a sigmoid function. First you don’t have a technology, then you have rapid progress, then the technology is kinda just “done” from a research perspective.

It’s possible that AI will be completely unlike previous technologies, and will be the first time we see something truly unique and self-perpetuating. But people have historically tended to overestimate how exponential things will be when trying to predict the future...

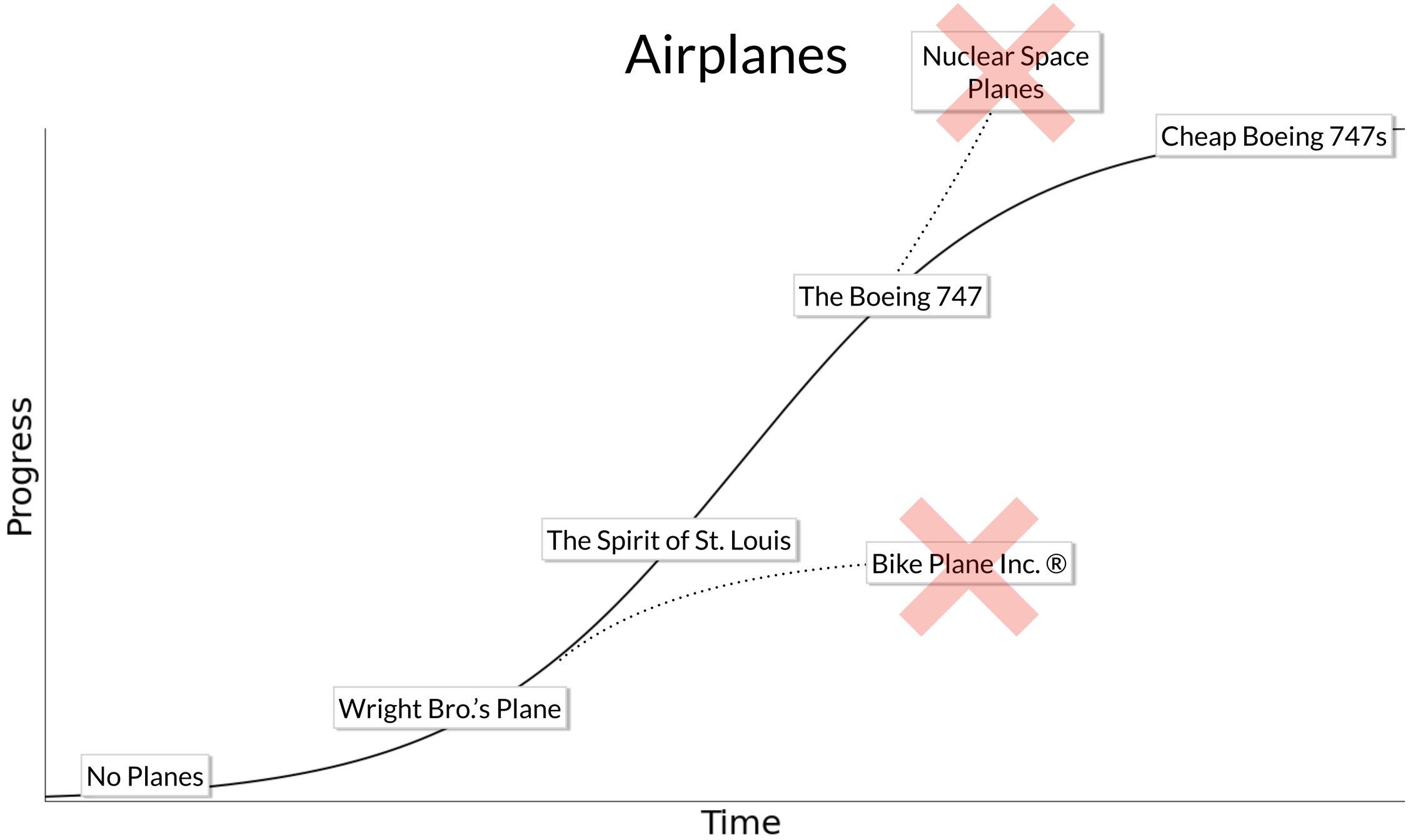


And you can see that sort of over-estimation here. This is a plot made by some scientists in the 1950s, where they are trying to predict how fast planes and rockets would be in their future. They were expecting we'd have interplanetary and interstellar spacecrafts by now, but of course, we don't.

Airplanes



Instead, airplanes followed a fairly boring trajectory. First we had no planes, then we had planes made out of wood and bike parts, then we had semi-modern planes, and then by 1970 we had the Boeing 747, which is a very good plane. And now 50 years later, we have cheaper and more fuel-efficient 747s. Airplane technology reached a certain level and then just saturated due to practical concerns: cost, noise pollution, convenience, lack of market demand, etc.



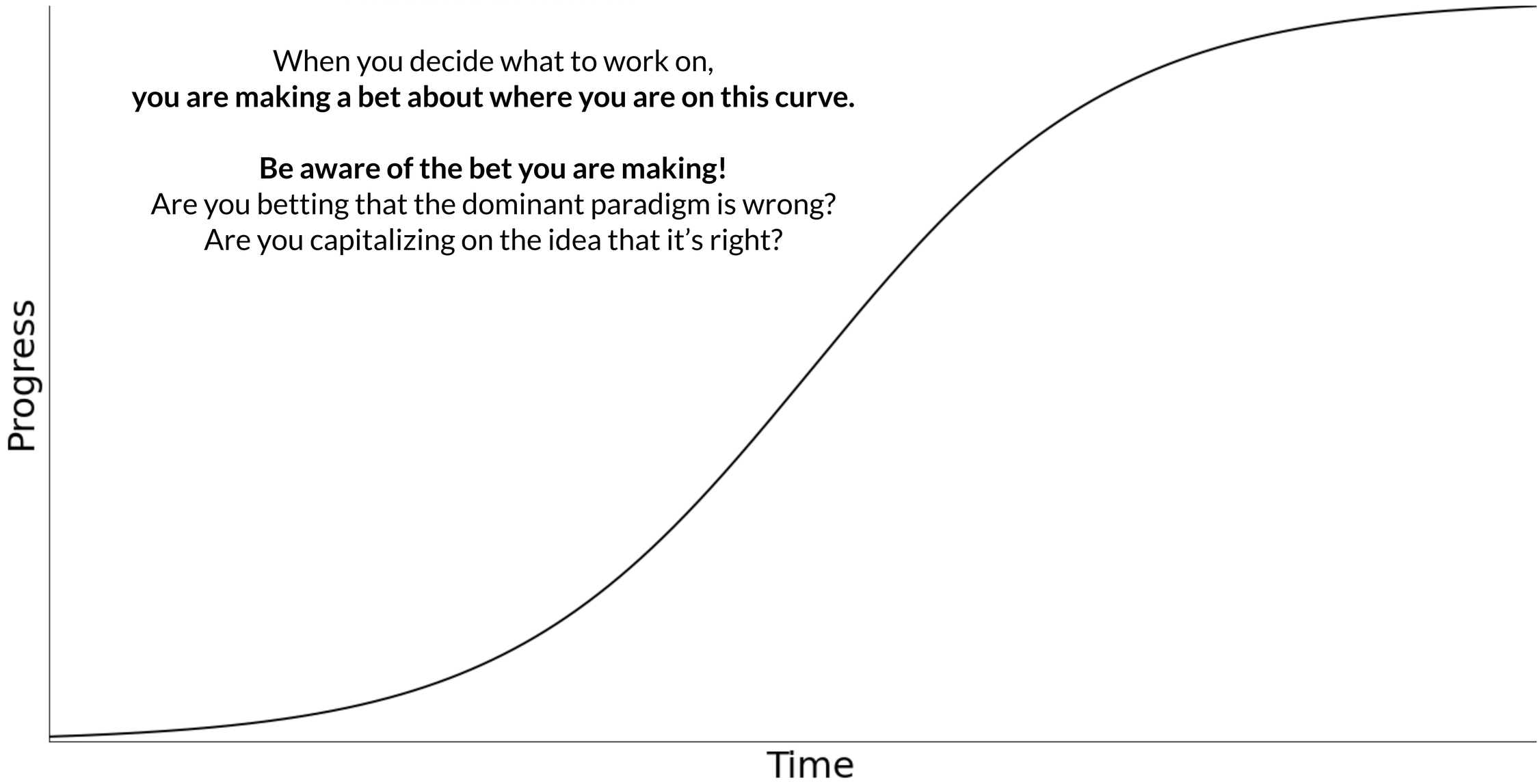
We never saw crazy nuclear space-planes, and if you were a researcher working on developing that sort of thing, your career likely suffered as a result. This is an error in judgement in terms of underestimating where we are on this sigmoid — maybe you thought we were just getting started with the Boeing 747, but as it turns out, that plane was basically good enough for everyone, forever.

Similarly, one could make an error in judgement in the opposite direction. Trying to commercialize the Wright Brother's plane would not have been a good idea.

The hardest problem a researcher has to solve is
“What should I work on?”

When you decide what to work on,
you are making a bet about where you are on this curve.

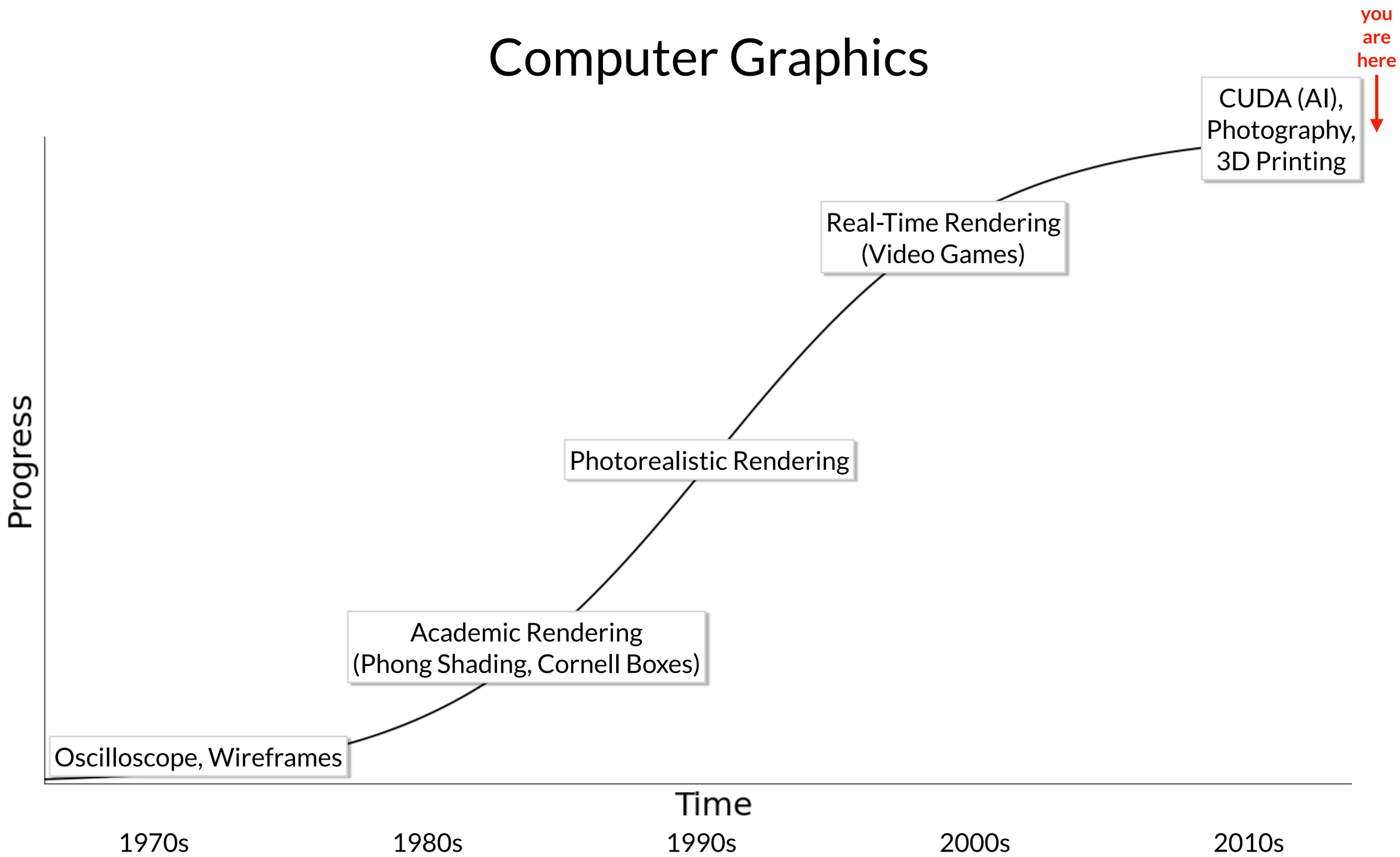
Be aware of the bet you are making!
Are you betting that the dominant paradigm is wrong?
Are you capitalizing on the idea that it’s right?



So you, as a researcher, have a very important meta-problem to solve, and it is “what should I work on?” This is an open problem! We don’t know the answer to it.

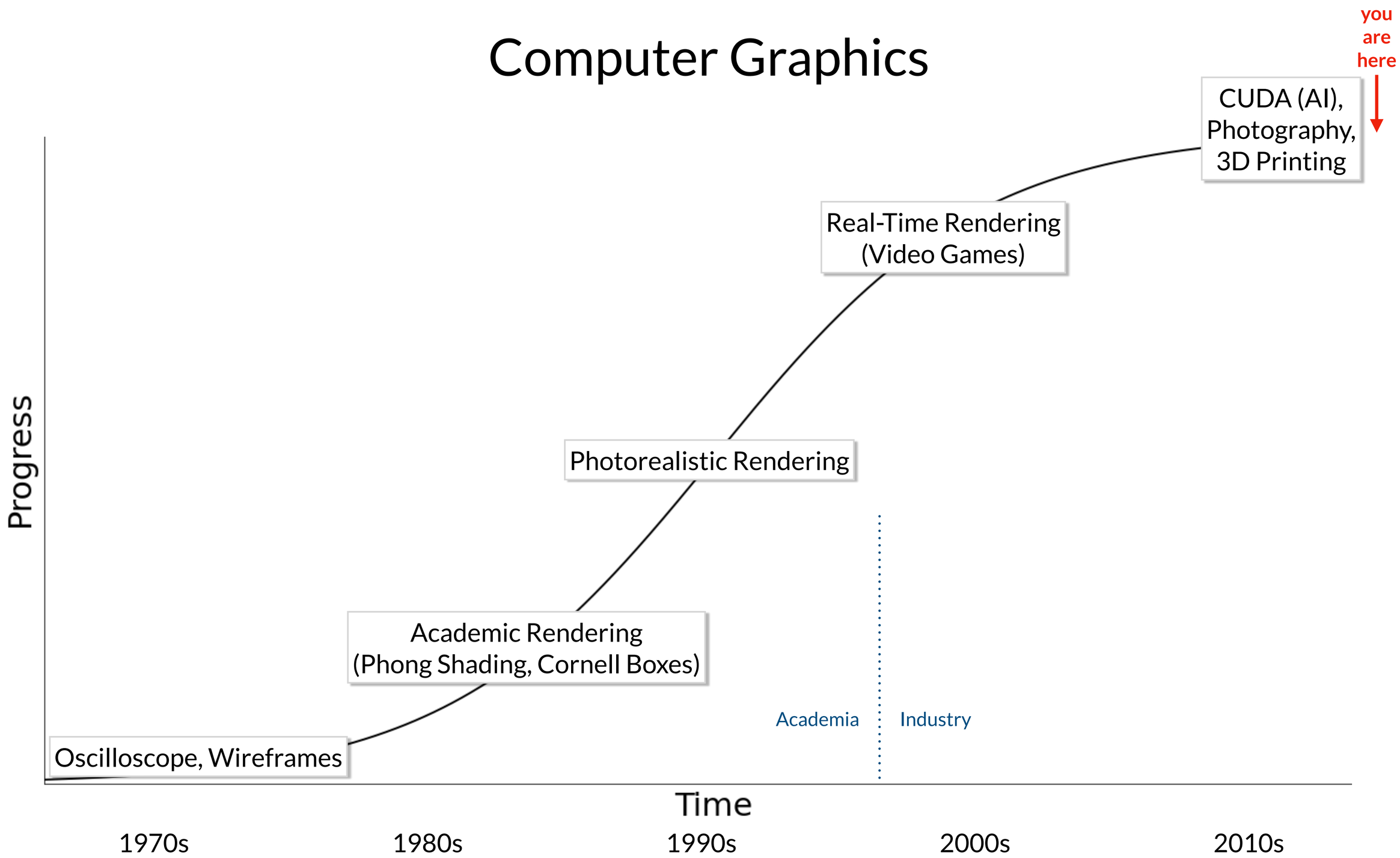
When trying to solve this meta-problem, you should try to figure out where you think we are on this curve. When you pick a research problem to work on, you are basically making a bet about where you think we are on the curve, and making a bad bet with your research career can be costly.

What sort of bet are you making? Do you think that we are far along on this sigmoid, and that the current paradigm is going to win out? Do you think we’re at the steep part? Or do you think that everyone is on the wrong path, and that we haven’t even reached the steep part?



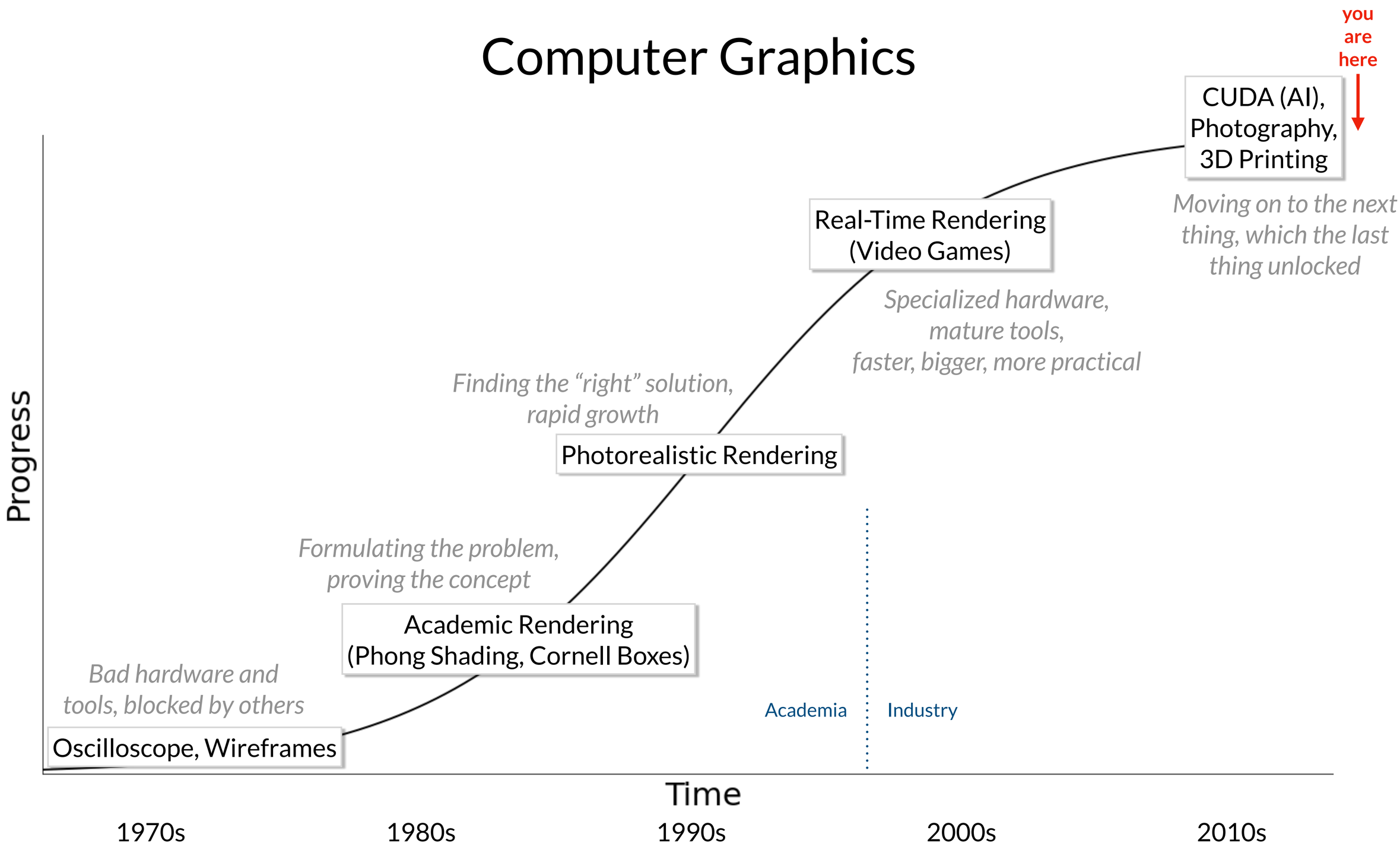
I don't know much about airplanes, but I know a little bit about computer graphics, because I've worked with lots of graphics people at Google. Here's my sketch of this sigmoid for graphics. People in vision like saying that graphics is "solved" as a research question, which is a bit of an overstatement, but we're definitely on the far end of the sigmoid.

Early attempts at graphics were fundamentally limited by inadequate hardware and tools, but people started figuring out what questions to ask and what solutions might look like. The central problem of the field was photorealistic rendering, and this was basically solved between the 80s and the 2000s. Then, the focus shifted towards making this research more broadly usable, which resulted in the rise of real-time graphics. Then the community ran out of central graphics questions, and pivoted to adjacent or successor technologies, like computational photography, or hardware and software tools for general purpose programming on graphics hardware. That latter bit ended up being **extremely important** for the broader field, as GPUs and CUDA were what unlocked deep learning, which unlocked the AI boom we are currently in.



Obviously this isn't a clear cutoff, but sometime in the 90s or 2000s the center of mass of the graphics community shifted from academia to industry. In the 80s the big things came from academia, like the rendering equation or phong shading, but by the 2000s it was places like Silicon Graphics and Pixar that were really moving the needle with hardware, software tools, and mass market media. This transition is a natural and positive part of the life-cycle of a technology — the University of Utah was not the right place to build mass market 3D graphics workstations, and the people building DirectX and OpenGL needed to have the math behind rendering nailed down before they could start building standards and toolsets.

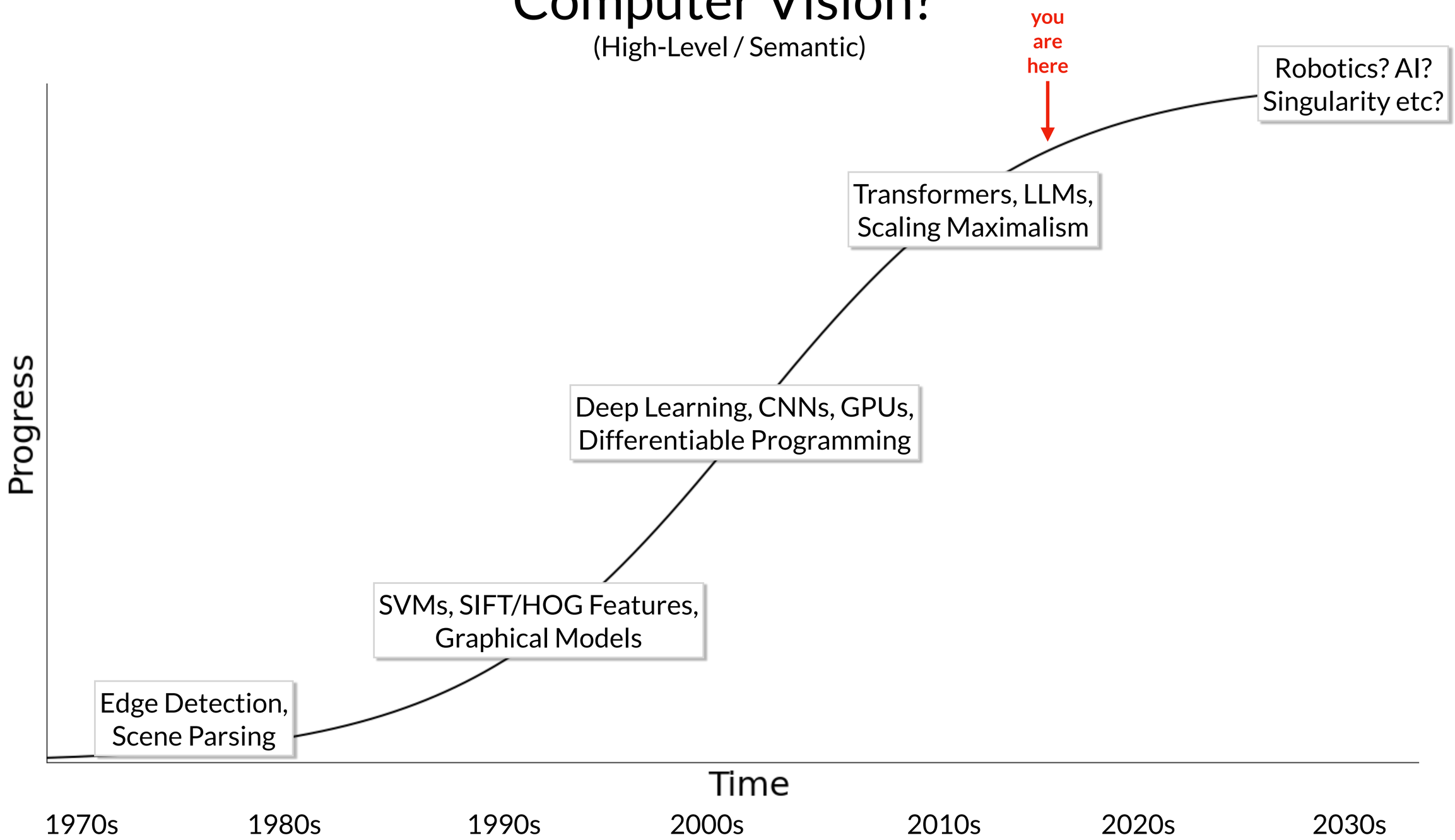
Obviously the academic graphics community continued to do well during and after this transition, but that community experienced two shifts: First, they started to become much more performance-oriented, because for their new algorithms to get industry adoption they needed to be **practical**. And second, they started to define their field much more broadly — things like 3D printing, computational photography, programming languages, and systems papers became mainstream at SIGGRAPH, when previously they would have been viewed as tangential or out of scope.



If you zoom out on these curves, you can start to draw more broad trends about what happens when. Early on, most research communities are just fundamentally blocked by their tools. Doing graphics with a 1970s workstation was just hard! You didn't have the compute or software resources you needed to execute everything well. Once people had the tools needed to make progress, they started figuring out what questions to ask and what answers they might have.

The middle of the sigmoid, where progress is steepest, is where people converge on the "right" question and the "right" answer. Things that previously seemed impossible (like having a film studio that used CGI for everything) become possible. Companies are formed and industry takes a bigger role. People start building specialized hardware and mature software stacks, and the focus shifts from "how do we do this" to "how do we do this more practically".

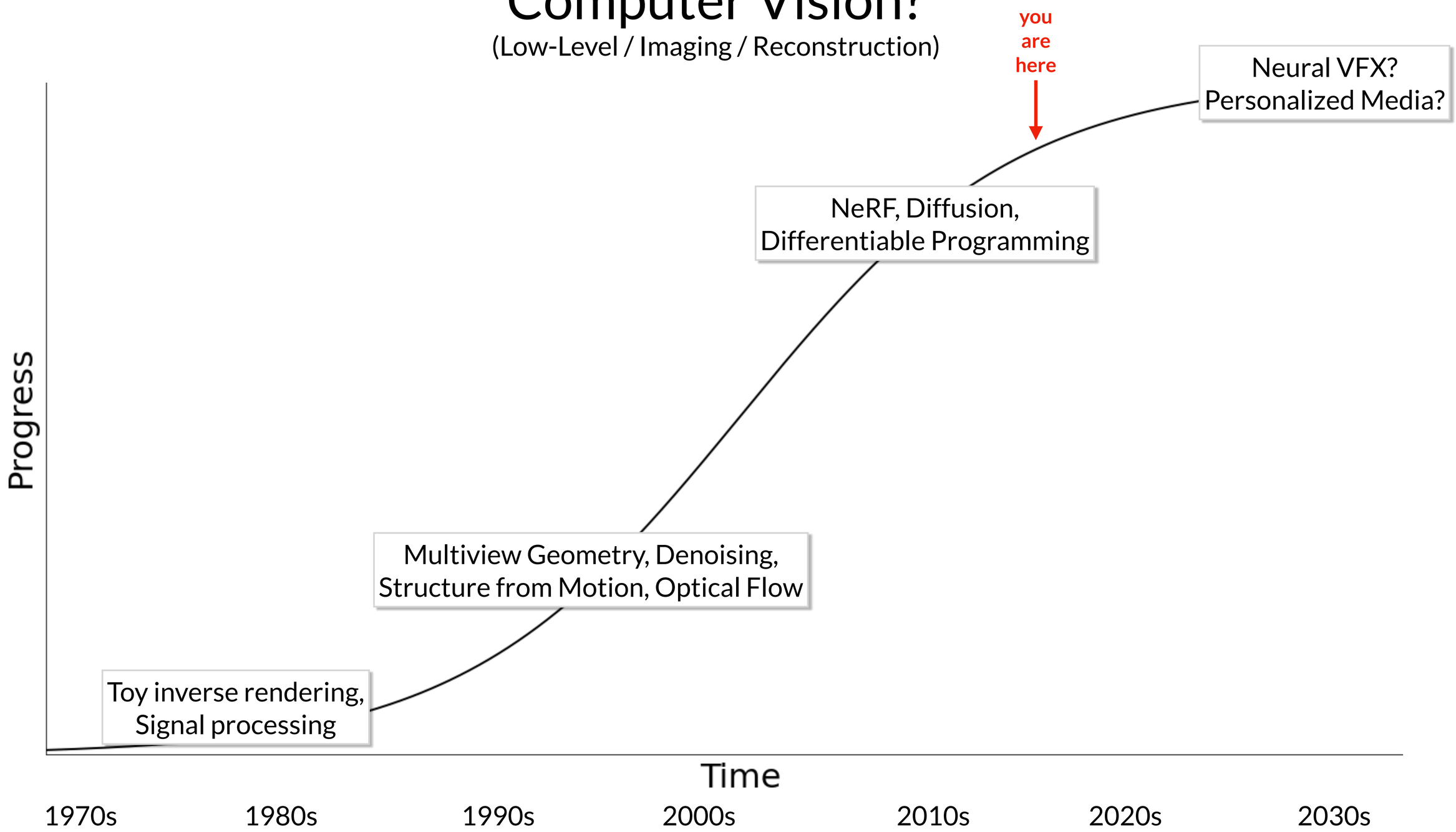
Once the field is very mature, the research community drifts away from what used to be the central problem of the field, and towards the **new research opportunities** that the field unlocked in its pursuit of the now-solved central problem.



Here's what this curve looks like for "high level" computer vision, in my humble opinion. We're not done, but we're close.

First we fumbled about in the 70s and 80s with models and problem statements that were a bad fit for the state of our tools (slow computers, LISP, symbolic logic). In the 90s and 2000s we started asking the right sort of questions and worked out first-order solutions to lots of those questions. The field hit its stride when deep learning became mainstream, largely because the ubiquity of GPUs and differentiable programming frameworks let researchers easily throw tons of compute at end-to-end optimization problems. Transformers and LLMs are a continuation of this paradigm; we have somewhat general-purpose differentiable computers and evidence that scaling maximalism is unusually effective for these problems, and I expect this trend will continue.

I'm not going to try to make firm predictions for what happens next, but I expect a lot of high-level vision people will shift towards things like perception for robotics, and I think this is going to be great for both the vision and robotics communities — after all, robotics used to be the justification for why any computer vision people in the 90s and 2000s would even consider working on object recognition.



And here's my estimate for what this curve look like for "low level" computer vision, which realistically is the only curve in this talk that I'm qualified to offer opinions about. We're also not done, but we're pretty far along. First we had bad signal processing on toy problems, and algorithms that never generalized outside of a handful of images. Then we nailed down denoising, geometry, and the basics of depth and motion estimation.

Then this field spun its wheels for a while, but started to see rapid progress again due to the rise of NeRF and diffusion-powered image synthesis techniques. These two advances are basically unrelated to each other, except that they are both downstream from the widespread adoption of differentiable programming and GPUs.

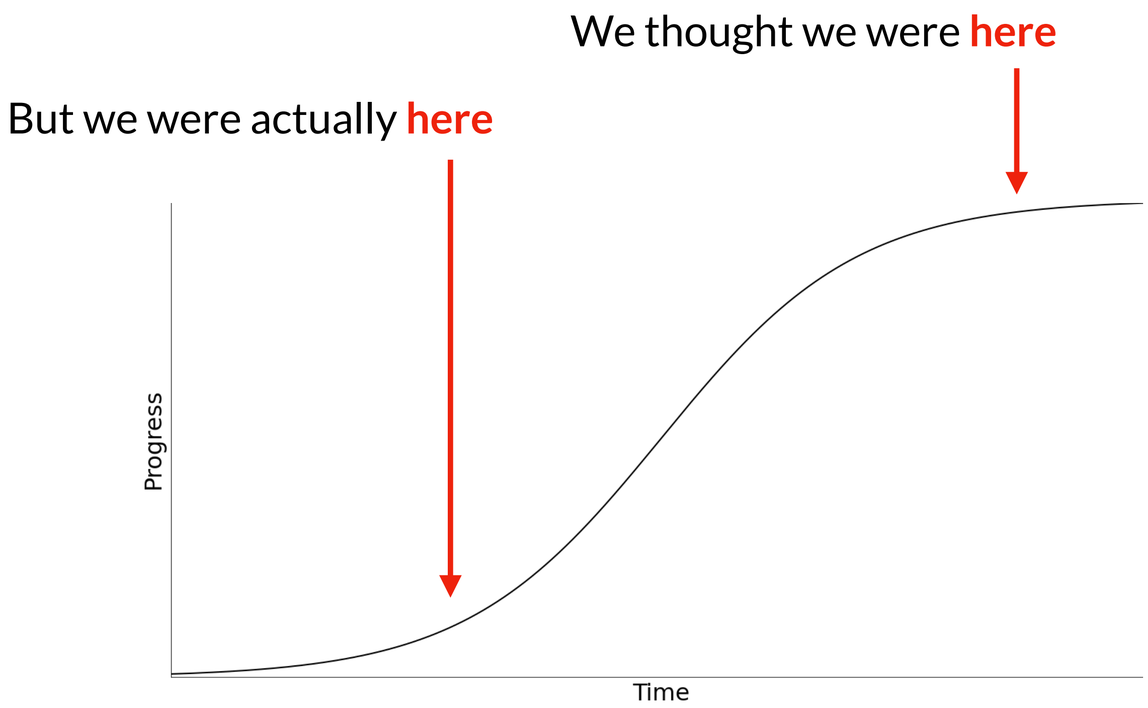
I don't know what's going to happen next, but these techniques will likely dramatically disrupt how all content creation works in the long term. This could be as straightforward as special effects studios and game devs using neural networks, or it could be a complete reinvention of all media into something that is fully generated and individually customized for each consumer.

Caveat: I am probably wrong

The computer vision community's **consensus view** circa ~2010 was:

- Working on denoising was pointless.
- 3D reconstruction was solved.

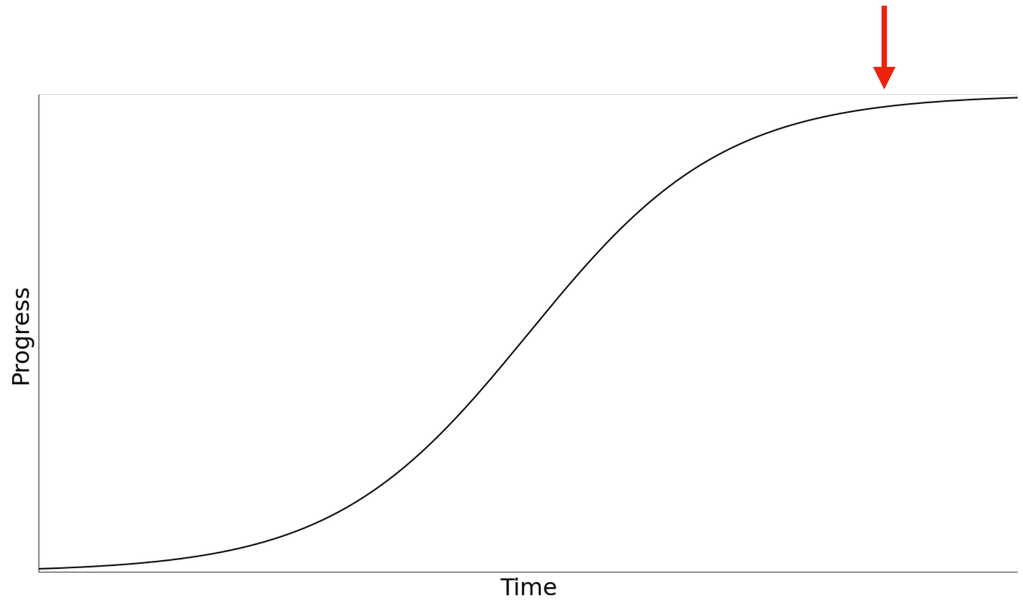
We were wrong! The rise of NeRF and diffusion were both a surprise.



All that said, you probably shouldn't trust what I'm saying! Making predictions about this stuff is really hard, and we've been wrong in the past. When I started working in computer vision ~15 years ago, it was universally accepted by almost everyone that working on low-level problems like denoising or 3D reconstruction was pointless, because both were completely solved problems.

But we were all wrong! It turns out that training CNNs to remove IID Gaussian noise was more useful than we expected, as this is the core underlying technology behind diffusion-based image synthesis algorithms. And though 2010-era 3D reconstruction systems were relatively mature and reliably produced reasonable results, they represented a local maximum that capped out at a quality level that was nowhere near what could be achieved through a different paradigm like NeRF. For both problems, we all thought that we were far to the right on the sigmoid, but we hadn't even really hit the steep part yet.

Some hard truths



- If we’re near the end of the sigmoid, industry is probably going to where the big things will happen.
- Industry research will probably get less research-y.
  Imagine Intel making chips, but it’s Google DeepMind and OpenAI fabricating PaLM-{n+1} and GPT-{n+1}
- Sometimes the “wrong” paradigm wins.
  In graphics, rasterization beat raytracing not because it was “right”, but because it had the most momentum (OpenGL, GPUs, etc)

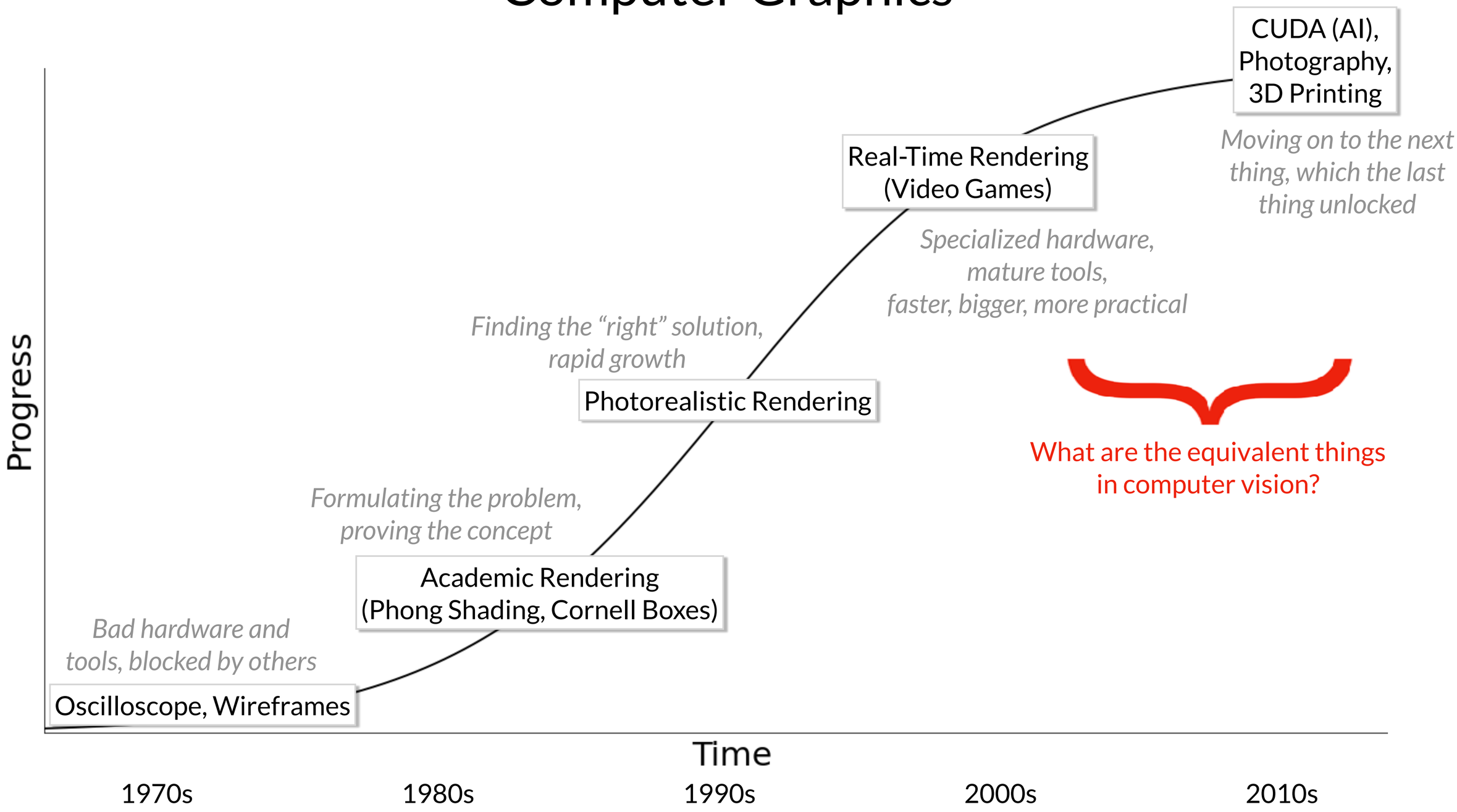
Okay, now it’s time for some hard truths about what being on the far end of this sigmoid means for researchers like us.

First, industry is probably where we should expect big things to happen. Industry is equipped to train the massive models that produce significant advances in capabilities, and academia is not. Maybe some unexpected paradigm shift will come along that lets people train SOTA models with severely limited compute resources, but I would not take that bet.

Second, we should expect industry research to become “less research-y”. We’re already seeing this: OpenAI functionally does not publish, and most large industry labs (except Meta) are increasingly reluctant to open-source their large models. Additionally, the roles of researchers within large industry labs will also probably start to change. Places like Google DeepMind and OpenAI may increasingly look more like the Apollo missions or Intel in the 80s — thousands of people involved in a concerted effort to accomplish the single goal of training and deploying the next giant model.

Lastly, you might not like the “foundation model” paradigm for AI, and that’s fine — there are many good reasons to believe that this paradigm is suboptimal or wrong in some way. But these objections might not matter much here. Very often, a paradigm wins not because it is “right”, but because it **already exists**. We are currently building massive hardware and software stacks around the “train a transformer on tons of data with tons of compute” paradigm, and this momentum gives an enormous advantage to that paradigm despite whatever flaws you may find with it. We saw this in computer graphics, where the rasterization-based pipeline that GPUs targeted won not because of its fundamental merits, but because if someone wanted to build a graphics system, they were at a tremendous disadvantage compared to their peers if they ignored the resources and effort that had been dumped into the dominant paradigm. If you tried to build a graphics engine while ignoring what GPUs and OpenGL can and can’t do, you’d have a hard time.

Computer Graphics

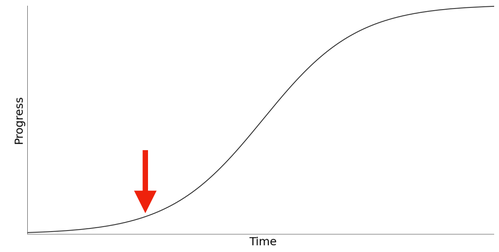


I showed this curve for computer graphics. What insights can we try to draw from it for computer vision? What will the end of our curve look like?

Moving on to the next thing, which the last thing unlocked

Robotics,
“Large World Models”,
3D recognition,
Video

Art,
Creative tools,
Custom media



The big thing here is probably “what’s next”? What are the new successor technologies that our current progress will unlock? For me, two big things stand out.

First, there’s still a ton of work to be done in terms of perception for robotics. Robotics is nowhere near the steep part of the sigmoid, and most of the robotics community is pursuing a paradigm for perception that doesn’t seem like it will pan out. This area is ripe for disruption. If we can figure out how to train an LLM-like model for perception in a dynamic 3D environment (a “large world model” maybe?), this will likely have a ChatGPT-sized impact on the world. And a lot of the technical problems here are likely well-suited to our community’s skill set.

The second huge opportunity here is AI-powered art and creativity. This isn’t exactly the computer vision community’s primary skill set, but we are well-positioned to have a huge impact in this space if we want to.

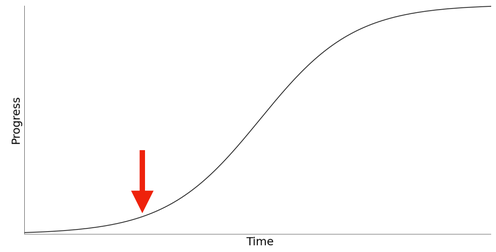
Credit: https://infinitevibes.net/

I feel a little crazy how under-appreciated this is in the mainstream vision and ML research communities, but the stuff we work on **accidentally spawned a new art form**, and nobody seems to be paying serious attention to it. The potential impact that all of this could have on the world is enormous, and it is bizarrely under-represented in our proceedings. There must be hundreds of interesting research questions in this space, and we should be the people identifying and answering those questions.

Moving on to the next thing, which the last thing unlocked

Robotics
“Large World Models”
3D recognition
Video

Art
Creative tools
Custom media



Specialized hardware,
mature tools,
faster, bigger, more practical

EFFICIENCY

The other big research opportunity I see is **efficiency**. This is less about “what’s next” and more about executing what we currently have more effectively.

The computer vision community seems to have a strong aversion to caring about efficiency, and this should change. In my experience, vision researchers tend to think that speed is somehow an intellectually “shallow” problem, and that it’s just an implementation detail. This is not the case! Thinking about how to make things faster is a deep and intellectually fulfilling problem, and it goes much further than just hand-optimizing CUDA. We are somewhat limited by our current toolkit here, as most researchers work by just re-arranging PyTorch modules, and it’s possible that making progress on this front may require a lower-level interface to our hardware. But I think the next generation of superstar vision and ML researchers (like the last generation of superstar graphics researchers) will be people who care about performance and speed, and are willing to think deeply about how their algorithm maps onto their hardware.

Efficiency

- Mainstream academic computer vision research has two main problems:
  - “How can I compete with well-funded industrial labs on benchmarks when scale always wins?”
  - “How can I meaningfully affect the trajectory of industrial research from outside?”

- Industry has a huge problem: most AI-powered solutions to problems are prohibitively expensive to deploy.

- Solution to all problems: Introduce benchmarks that **NORMALIZING BY COST**, by measuring accuracy…
  - Per watt
  - Within some constrained model size
  - Within some finite compute budget

- The computer vision community doesn’t like thinking about speed and efficiency, and this has to change.
  - Industry will always win at $\mathrm{maximize}(\mathrm{accuracy})$
  - Everyone has an even playing field if we $\mathrm{maximize}(\mathrm{accuracy}/\mathrm{watts})$

The way I see it, academics in computer vision have two huge problems: how to compete with industry on benchmarks, and how to meaningfully affect industry research from the outside. But industry also has a critical problem: most AI-powered solutions to problems are **too expensive to ship**. We need multiple order of magnitude speedups to our current techniques to make them broadly applicable in all settings, and this problem isn’t going to be solved through hardware improvements — we’re bottlenecked by the laws of physics, and we need better algorithms.

All of these problems have a single tidy solution: the CVPR community should shift towards benchmarks that **normalize by cost**. We shouldn’t compare papers in terms of just accuracy, we need to consider speed, model size, energy consumption, and other practical factors. If we only care about accuracy in a vacuum, industry will always win on benchmark leaderboards, and academics will struggle to have impact. But if we nudge the research community towards caring about more holistic metrics, 1) we’ll even the playing field between industry and academia, and 2) we’ll encourage research breakthroughs that actually move the needle for deployed industry models, which is a win for everyone.

Thanks


jonbarron@gmail.com


@jon_barron


http://jonbarron.info/

Thanks!