Essay_v1

Anton Novokhatskyi

Robotics is about to change. Not eventually, soon. I've spent the last two years building in this space, and what I'm seeing now is different from anything before.

It started as a feeling I couldn't quite pin down. I'd be training a policy on a Tuesday night, and something that failed for months would just... work. Then I'd read a paper that solved a problem I assumed was years away. Then a $300 robot arm would show up that actually performed. These weren't isolated events. They kept happening, faster and faster, until I stopped being surprised and started paying serious attention.

What's changed isn't one thing. It's everything moving at once. Models that understand 3D space. Simulation that's finally good enough to transfer to real hardware. Robots you can actually afford to break while learning. Two years ago I was stitching together brittle pipelines and hoping for the best. Now I'm watching foundation models control physical arms with the kind of fluency that LLMs brought to text in 2022, and I remember what that felt like from the outside. This feels the same, except I'm inside it.

I'm not writing this as someone who read about the future of robotics. I'm writing it as someone who's been up at 3am debugging sim-to-real transfer, who has trained models that did nothing for weeks and then suddenly did something extraordinary. I have opinions because I've earned them the hard way. This essay is about what I've seen, what I'm building, and why I think anyone paying attention should be building too.

The Data Problem

Every AI breakthrough follows the same pattern. Stop writing rules by hand, feed the model enough data, and watch capability emerge. NLP had decades of hand-crafted grammars. All of it became irrelevant the moment we could train on the internet. Computer vision had SIFT, HOG, years of clever feature engineering, gone once we had ImageNet and the compute to chew through it. The lesson is always the same: data at scale is what unlocks everything.

Robotics doesn't have this. There's no internet of robot demonstrations. No massive repository of physical interactions waiting to be downloaded. What exists are fragments: a few thousand trajectories recorded in specific labs, on specific robots, for specific tasks. Human videos exist at scale, but they don't map to robot embodiments. Nothing composes into the kind of dataset that taught GPT how language works or taught CLIP how images relate to words.

So we're stuck with behavioral cloning. You physically demonstrate a task, record the trajectory, and train a policy to replay what it saw. It works, but only in the narrowest sense. The robot learns that grasp, at that angle, on that object. Shift anything (a different tomato, a slightly rotated bin, light coming from the other side) and the policy breaks. It memorized the demonstration without understanding the physics underneath it. BC doesn't generalize because it was never given enough data to learn what generalizing even means.

And scaling past this is brutal, because the bottleneck is human time. Every additional demonstration requires a person physically moving a robot arm through a trajectory. Compute has gotten absurdly cheap. Storage is practically free. But a human collecting demonstrations at a client site costs the same per hour it always did. In a field where everything else has gotten faster, human data collection is the constraint that refuses to move.

I keep thinking about it this way: the hardware is the body (motors, joints, sensors, the physical thing that moves through space). The algorithms are the nervous system; they process signals and produce actions. But the thing that makes it intelligent, the thing that gives a robot the ability to understand how objects behave when you push them, grip them, stack them, that comes from data. Data is what contains compressed knowledge of how the physical world works. We've spent years building better bodies and refining the nervous system. What's missing is the thing that brings it alive.

We have spent years building better bodies. We have spent years refining the algorithms. What we lack is the soul.

A Lesson from Biology

For a long time, I was drawn to biology and neuroscience not because I wanted to study living things, but because I was obsessed with a question: how does information become behavior? At some point I stopped seeing the world as made of objects and started seeing it as made of signals. Light hitting retinas, pressure on nerve endings, electrical impulses racing through tissue. Everything I experience is just my brain running algorithms on sensory input. Once you see it that way, you can't unsee it. And it changes how you think about machines.

The idea that reshaped my thinking about AI systems comes from evolutionary biology. Dawkins argued in The Selfish Gene that organisms are temporary vehicles for genetic information. Genes are the real replicator, and bodies are just their survival machines. For three billion years, this was the only game in town. Evolution was slow, physical, measured in millions of years.

Then, roughly 2.5 million years ago, something new emerged. The meme, a unit of information that spreads through imitation rather than inheritance. Where genes replicate through DNA, memes replicate through brains. And humans turned out to be extraordinarily good at this, by following the evolution. Imitation is a super-skill: if you can copy the making of a stone axe, you can copy fire-making, medicine, language, technology. Humans became machines for super-imitation.

But here is the insight that matters: a human raised in complete isolation would be an empty biological shell. Genetically human, yes, but not functionally human. The genes build the hardware; the memes provide the software. Neither is sufficient alone. More importantly, they co-evolved. Memes created selective pressure for bigger brains; bigger brains enabled more complex memes; this feedback loop produced everything we recognize as human intelligence.

I see the exact same architecture in robotics.

Pre-training a robot through behavioral cloning is genetic evolution. It builds a general-purpose "copying machine," a model that understands how to move, how to perceive, how to act in broad terms. But this pre-trained model, without task-specific learning, is like a human raised in isolation. It has the architecture but lacks the skills. It is genetically complete and memetically empty.

Reinforcement learning is the memetic evolution, the second replicator. It takes the general-purpose substrate that pre-training created and transmits specialized skills into it. It does not build the body from scratch; it teaches the body what to do. And just as memetic evolution operates orders of magnitude faster than genetic evolution, RL in simulation operates orders of magnitude faster than real-world data collection.

The simulation environment I am building is the primeval soup, the medium in which this second evolution can take place.

3D Digital Worlds

But how do we create these simulation environments at scale? The data already exists. Every factory floor photographed for documentation, every warehouse walkthrough filmed on a phone, every client site captured during an initial visit. The missing piece was never the data. It was the ability to turn that data into worlds. 3D reconstruction techniques are beginning to close that gap, converting photos and videos into spatial environments that a robot can train inside. Reconstructions derived from reality, not artist-built approximations of it.

The moment this stopped being abstract for me was a barbecue at my yard. Friends visiting, good light, one of those evenings that just works. We discussed the Gaussian Splatting together and we decided to try it out. I captured the scene on a video, no tripod, no plan, just my phone and me going around the tasty looking barbecue and capturing it from all angles. When I ran the reconstruction later and shared it with people, something unexpected happened. They didn't say "that looks nice." They felt like they were there. That reaction told me something important. A photograph shows you a scene. A spatial reconstruction gives you the scene, the feeling of being there: the way light falls across surfaces, how objects sit in relation to each other, the implicit geometry of a space you can move through. The information encoded in three dimensions is orders of magnitude richer than anything flat media captures. If we can reconstruct real environments with that fidelity, we can give machines something to reason about that actually resembles the world they need to operate in.

The mathematical elegance of Gaussian splatting is what makes this practical. A Gaussian distribution is perhaps the most natural object in all of machine learning. It appears everywhere, from probability theory to neural network activations to the fundamental assumptions underlying how we model uncertainty. Representing 3D scenes as collections of Gaussians means working with continuous, differentiable structures that plug directly into everything we already know about training neural networks. Discrete representations were designed for rendering pipelines built decades ago. Continuous representations are designed for learning.

The research trajectory confirms what the mathematics suggests. Publications on Gaussian splatting have grown dramatically over the past four years. Marble Labs is building generative models that create 3D worlds from scratch. Meta's SAM3D extends scene understanding into three dimensions and lots of other researches are happening in this field. The community is not just exploring this direction, it is committing to it. There are real problems still unsolved: physics estimation, collision detection, interaction modeling on Gaussian representations remain genuinely difficult. But these are problems of engineering and scale, not fundamental impossibilities. The fact that talented researchers are actively targeting them gives me confidence that solutions will emerge.

Video-based world models like Google's Genie or NVIDIA's generative simulators take a different approach, learning physics directly from visual sequences rather than reconstructing explicit geometry. I believe both directions will contribute meaningfully to how machines understand and interact with the world. We need people working deeply in each. My own conviction, shaped by years of working with 3D and that visceral barbecue moment, pulls me toward explicit spatial representations.

What gives me additional confidence is how the industry is evolving. NVIDIA's strategy is instructive: they build the hardware, then give away the tools (Isaac Sim, Cosmos, Nurec, Omniverse, CUDA ecosystems) freely to anyone on their platform. This is not charity; it is ecosystem building done right. It means the barrier to entry for simulation-based robotics research is lower than it has ever been, and collaboration across institutions becomes natural rather than negotiated. When the infrastructure layer is open, innovation happens faster. That is the environment I want to work in, and that is why now is the right time.

What I Am Building

The gap between a beautiful reconstruction and a useful simulation is enormous. I know because I've been living in it.

The first time I loaded a Gaussian splat into a physics engine and dropped a virtual object onto the surface, it fell straight through. The reconstruction looked perfect, every texture, every shadow in the right place, but there was nothing there. No collision surface, no friction, no mass. A robot training in this space would learn to reach for objects it could never touch. I had built a painting, not a world.

Closing that gap is what I work on now. The goal is a pipeline that takes a camera scan of any real environment and produces something a robot can physically interact with. Push, grasp, knock over, and learn from. That means extracting collision geometry from the reconstruction. It means estimating physics properties: which surfaces are slippery, which objects are heavy, which things deform when you squeeze them. It means being able to segment individual objects out of a scene so they can be manipulated independently. And it means generating variations (different object positions, different lighting, different sizes) so the robot doesn't memorize one arrangement but learns the task itself.

Every one of these pieces exists as isolated research. Semantic segmentation of Gaussian splats. Differentiable mesh extraction. Vision-language models that estimate material properties from images. Generative 3D models that create objects from text. What nobody has done is connect them into a single automated system and ask: is the result good enough to train a manipulation policy that transfers to a real robot?

That's the question I'm trying to answer. On a specific task, with a specific robot, measuring specific numbers. But the question behind the question is bigger: can we stop hand-building simulation environments entirely? Right now, creating a training scene for a new task takes weeks of manual work. If an automated pipeline can produce something good enough, the economics of the entire field change. You don't need an army of simulation engineers. You need a camera and a few minutes.

And if you can generate one scene, you can generate thousands. Vary the lighting. Randomize object positions. Swap materials. Generate new objects that don't exist in the original scan. Each variation is another training episode. Each episode teaches the policy something slightly different about how the world behaves. This is where the biological metaphor becomes literal: the simulation becomes an environment where policies evolve, where the selection pressure of millions of trials shapes behavior that no amount of human demonstration could produce.

This is what continuous learning looks like. A system that keeps learning, in simulation, at scale, across an infinite space of scenarios that would be impossible to construct in the physical world. A tomato sorting robot that has seen every possible tomato, in every possible position, under every possible lighting condition, before it ever touches a real one. The real world becomes the final exam, not the classroom.

Closing

I don't know if this pipeline will work. I know the meshes I extract from Gaussian splats sometimes shatter during physics simulation. I know VLM-estimated friction values might be wildly off. I know that a policy trained in a beautiful reconstruction might still fail the moment it sees a real tomato under fluorescent light. These are open questions, and they have answers I haven't found yet.

But I'd rather be the one finding out than the one waiting for someone else's paper. The field is moving fast enough that the interesting conversations are happening between people who are building, not between people who are speculating. I want to find those people. I want to hear what they've tried, what surprised them, what broke in ways they didn't expect.