MY WEIRD PROMPTS

Podcast Transcript

EPISODE #210

Predictive Motion: How Transformers Are Learning to Walk

Published January 09, 2026 • Runtime: 23:04

https://myweirdprompts.com/episode/embodied-ai-robotics-transformers/

EPISODE SYNOPSIS

In this deep dive, Herman and Corn explore the radical convergence of large language models and robotics, marking a transition from digital logic to physical embodiment. They break down the mechanics of Vision-Language-Action (VLA) models, explaining how the transformer architecture is being repurposed to predict motor commands just as it predicts words. By treating physical movements as "action tokens," researchers are bridging the gap between abstract reasoning and real-world coordination. The discussion covers the critical "reality gap," the role of high-fidelity simulations like NVIDIA Isaac Sim, and the necessity of low-latency edge computing for the next generation of humanoid robots. Whether it’s a robot arm grasping a cup or a humanoid navigating a kitchen, the duo questions if true intelligence can only be achieved when AI finally has a body to call its own.

DANIEL'S PROMPT

Daniel

We’ve talked a lot about the mechanics of AI—such as transformer architecture, predictive reasoning, and vector matching—but I’m curious about embodied AI. To what extent are these fundamental mechanics relevant to robots taking action in the physical world? Are models for synthesizing intelligence in robotic systems a totally different breed, or do the principles we’ve discussed so far also apply to embodied AI?

TRANSCRIPT

Corn

So, Herman, I was watching our housemate Daniel try to assemble that new flat-pack bookshelf in the living room yesterday. It was a fascinating exercise in what we might call embodied intelligence, or perhaps in his case, a lack thereof at three in the afternoon. He had the instructions, he had the parts, but the actual physical coordination, the spatial reasoning, the way he had to adjust his grip when a screw wouldn't quite catch—that's something we don't usually talk about when we're diving into the guts of large language models.

Herman

Herman Poppleberry here, and I have to say, I was watching him too, mostly because I was worried about the hardwood floor. But you're right, Corn. Daniel actually sent us a prompt about exactly this. He noticed that while we spend a lot of time talking about transformers, predictive reasoning, and vector matching in the digital realm, he's seeing more and more about embodied A I lately. He specifically mentioned seeing those new Gemini-powered multimodal and agentic A P Is increasingly being used for robotics in major model dashboards. He wanted to know if the stuff we've been discussing for the last two hundred episodes actually applies to robots, or if we're looking at a completely different breed of intelligence when a machine has to actually move through the world.

Corn

It's a great question because for a long time, robotics and artificial intelligence were almost like two separate silos. You had the A I researchers working on logic and language, and the roboticists working on control theory and kinematics. But here we are in January of twenty twenty-six, and those silos have basically collapsed into each other. I think a lot of people assume that a robot is just a computer with limbs attached, but the reality of how these models are being synthesized is much more nuanced. Are the fundamental mechanics the same? Or does the physical world force us to throw out the transformer architecture and start over?

Herman

That's the meat of it, isn't it? To answer Daniel's question directly: the principles are surprisingly similar, but the implementation is where it gets wild. If you look at the state of the art right now—things like recent open vision-language-action research models and the foundation models running on the latest Figure and Tesla hardware—we're seeing the rise of what researchers call V L A models. That stands for Vision-Language-Action models. And at their core, they are still transformers. Remember back in episode one hundred and five when we talked about A I benchmarks and how models predict the next token? Well, in embodied A I, the robot is still predicting the next token. It's just that the token isn't a word. It's a motor command, or what we call an action token.

Corn

Okay, let's pause there because that's a huge conceptual leap for most people. When I type a prompt into a chatbot, the token might be the word apple. When a robot is looking at an actual apple on a table, how do you turn a physical reach-and-grasp movement into a token?

Herman

It's all about discretization. Think about how we turned images into tokens. We don't feed a model a raw stream of pixels anymore; we break the image into small patches, and each patch becomes a vector that the transformer can process. In robotics, we do something similar with actions. We take the range of motion of, say, a seven-degree-of-freedom robotic arm. We discretize those movements into a finite set of possibilities. So, instead of an infinite range of motion, the model sees a library of micro-movements. Moving the wrist three degrees to the left might be token number five thousand four hundred and twenty-two.

Corn

So, the transformer is essentially playing a very high-stakes game of Mad Libs with a robot's joints? It's looking at the visual input, looking at the command like pick up the red cup, and then predicting the sequence of motor tokens that results in the cup being lifted?

Herman

Exactly! And this is where the predictive reasoning we've talked about becomes so powerful. In a standard language model, the context window holds the previous words in the sentence. In an embodied model, the context window holds the previous few seconds of video frames and the previous motor commands. The model is constantly asking, based on what I just saw and what I just did, what is the most statistically likely next movement to achieve the goal? We call this action chunking—predicting a whole sequence of movements at once so the robot doesn't look like it's glitching every millisecond.

Corn

That's fascinating, but it also sounds terrifyingly fragile. If I'm writing a poem and the model picks a slightly weird word, the poem is just a bit avant-garde. If a humanoid robot is carrying a tray of glasses and its next-token prediction is off by five centimeters, we have a disaster. How do these models handle the sheer unforgiving nature of physics compared to the relative safety of a text box?

Herman

You've hit on the biggest point of friction between digital A I and embodied A I. It's what we call the reality gap. In the digital world, data is cheap and mistakes are free. In the physical world, data is expensive because you have to actually run the robot, and mistakes involve broken hardware or, heaven forbid, injured humans. This is why vector matching is so critical here. When a robot encounters a situation it hasn't seen before, it uses those high-dimensional embeddings to find the nearest neighbor in its training data. It's looking for a functional similarity. It's thinking, okay, this blue mug is shaped differently than the white one I trained on, but in vector space, their grasp points are very close.

Corn

I remember we touched on this a bit in episode two hundred and eight when we were talking about satellite intelligence—how the model has to recognize patterns across different lighting conditions and angles. But with a robot, there's an extra layer: proprioception. It's not just seeing the world; it has to sense itself. Does a transformer naturally handle the internal state of a robot, or do we need a separate system for that?

Herman

That's one of the coolest developments of the last year. Many modern embodied models can treat proprioceptive data—the feedback from sensors in the joints telling the robot where its arm is—as another input stream alongside vision and language, often encoded into vectors or tokens and processed by a shared model. The transformer architecture is brilliant at finding the relationships between these different streams. It learns that when the visual tokens show the hand touching a table, the pressure sensor signals should spike. If they don't, the model realizes something is wrong. It's integrating vision, touch, and balance into a single latent space.

Corn

So, the transformer is acting as the central nervous system, integrating all these senses into a single world model. But here's where I want to push back a little. We've talked before about how L L Ms don't really have a sense of cause and effect; they just have statistical correlations. In the physical world, cause and effect are everything. If I push a ball, it rolls. If a robot is just using statistical patterns to move, does it actually understand that it's the cause of the ball moving? Or is it just predicting that the next frame of video will show a rolling ball?

Herman

That is the million-dollar question in twenty twenty-six. Some researchers argue that true understanding only comes from embodiment. They call it the physical grounding of intelligence. There's this idea that you can't truly understand the word heavy until you've tried to lift something that exceeds your motor torque. When these models are trained on physical data, they start to develop an implicit physics engine. They aren't calculating Newton's laws with equations; they're intuiting them through billions of frames of experience. It's very similar to how you and I move. You don't calculate the parabola of a falling ball to catch it; your brain has a predictive model of where that ball will be.

Corn

It's like the difference between knowing the dictionary definition of gravity and feeling the weight of a heavy box. But this leads to a massive bottleneck: data. We have the entire internet to train language models. Where do we get the data to train a robot to do a trillion different physical tasks? We can't just let a million robots wander around breaking things for a decade.

Herman

You're right, the data bottleneck is the primary reason why your kitchen isn't currently being cleaned by a robot while we're recording this. There are three main ways we're solving this right now. First, there's teleoperation. Humans wear V R suits and perform tasks, and the robot records every movement and every visual frame. This is high-quality data, but it's slow to collect. Second, there's video pre-training. We feed the models millions of hours of YouTube videos of people doing things—cooking, cleaning, repairing cars. The model learns to associate visual sequences with tasks.

Corn

Wait, can a model really learn to move just by watching humans? Humans have different proportions, different muscle structures, and frankly, we move a lot more fluidly than most current hardware.

Herman

It's called cross-embodiment learning. The model learns the high-level strategy from the human video—like, first you pick up the knife, then you hold the onion, then you slice. Then, it uses a smaller set of robot-specific data to translate that strategy into its own motor tokens. But the third way, and this is the one that's really exploded in the last eighteen months, is simulation. We've gotten incredibly good at creating photo-realistic, physics-accurate digital twins of the world using platforms like NVIDIA Isaac Sim and research frameworks such as Isaac Lab. We can run a thousand simulations in parallel, where a virtual robot practices opening a door ten million times in a single afternoon using something called domain randomization—basically making the virtual world slightly messy so the robot is prepared for the chaos of the real world.

Corn

And then you just... copy-paste that brain into the physical robot? Is it really that simple?

Herman

Not quite. There's still that sim-to-real gap. A simulation can never perfectly capture the friction of a specific carpet or the way a hinge might be slightly rusty. This is where the predictive reasoning needs to be adaptive. The model has to be able to say, in the simulation, this door took five Newtons of force to open, but in the real world, it's taking seven. I need to adjust my next-token prediction in real-time. This is why low latency is so much more important in robotics than in a chatbot. If your chatbot takes two seconds to respond, it's annoying. If your robot takes two seconds to respond to a slip, it falls over.

Corn

That brings up a technical point I've been curious about. In our previous discussions, like back in episode two hundred and five when we talked about managing massive data, we focused on throughput. But in robotics, we're talking about inference speed at the edge. Are these robots carrying a massive G P U rack on their backs, or are they beaming their thoughts to a cloud server? Because if Daniel's kitchen robot has to wait for a round-trip to a data center in Virginia just to catch a falling egg, that egg is toast.

Herman

Most of the sophisticated humanoid robots we're seeing in twenty twenty-six, like the latest Digit or Tesla's Optimus prototypes, are using a hybrid approach. They have a powerful edge-computing unit—often a specialized A I chip similar to platforms like the Jetson Thor—for the immediate, low-level reactive stuff. Balancing, obstacle avoidance, basic grip adjustments. That has to happen locally with sub-ten-millisecond latency. But the high-level reasoning—the part that says, I should probably put the heavy groceries at the bottom of the bag—that can be handled by a larger model, sometimes even in the cloud, because it doesn't need that instant feedback loop.

Corn

It's like the human nervous system. Your spinal cord handles the reflex of pulling your hand away from a hot stove before your brain even realizes you've been burned. The high-level transformer is the brain, and the low-level controllers are the spinal cord. But Daniel's prompt also asked if these are a totally different breed of model. From what you're saying, it sounds like they're more like a specialized evolution of the same breed.

Herman

Exactly. It's the same fundamental architecture—the attention mechanism, the transformer blocks, the embedding spaces—but the training objective is different. Instead of predicting the next word in a sentence, the objective is to minimize the error between the predicted state and the actual state. When we talk about predictive reasoning in an L L M, we're talking about semantic space. In robotics, we're talking about state space.

Corn

Let's dive into that state space idea. If I'm a robot, my state is my position, my velocity, the torque on my joints, and the visual field in front of me. That's a huge amount of data to compress into a vector. How do these models decide what's important? Like, if I'm a robot trying to pour a glass of water, the color of the wall behind the glass shouldn't matter, but the angle of the pitcher is everything. Does the attention mechanism handle that automatically, or do we have to hard-code what the robot should pay attention to?

Herman

That's the beauty of the attention mechanism. It's learned. During training, the model realizes that when the goal is pour water, the attention weights on the pixels representing the pitcher's rim and the glass's opening become very high, while the weights on the background noise drop to near zero. This is what we call visual attention. It's the same thing that allows a language model to realize that in the sentence The bank was closed because the river overflowed, the word bank refers to the land, not a financial institution. The context dictates the attention. In robotics, the task dictates the attention.

Corn

That's a great analogy. It makes the whole thing feel much less like magic and much more like the statistical processing we're used to. But there's a misconception I want us to address, because I hear this a lot. People think that to make a robot smart, you just have to give it a better set of instructions. Like, step one: move hand to coordinate X, Y, Z. Step two: close grip. But what you're describing is a system that doesn't really have steps in the traditional sense. It has a continuous flow of probability.

Herman

You've hit on the biggest shift in robotics in the last five years. We've moved from imperative programming to end-to-end learning. In the old days, if you wanted a robot to pick up a cup, you had to write thousands of lines of code covering every possible angle and light condition. It was brittle. If the cup moved an inch, the code failed. Today, we use end-to-end neural networks. You feed the model the camera feed and the goal, and the model outputs the motor commands directly. No middleman, no pre-defined steps. It's much more robust because it's learned to generalize. It's not looking for a specific coordinate; it's looking for the concept of a cup in its visual-vector space and the concept of a grasp in its action-vector space.

Corn

So, when Daniel looks at that Gemini A P I activity around robotics, he's basically looking at a way to give a robot a high-level brain that already understands the world. But here's the kicker: hallucination. We know that transformers hallucinate. They make things up. If a robot hallucinations a third arm or thinks the floor is a table, that's a serious problem. How do we build safety into a system that is fundamentally based on probability rather than logic?

Herman

This is the frontier of research right now. We're seeing the implementation of what are called constrained policy layers or safety shielding. Basically, the transformer suggests an action, but before that action is sent to the motors, it passes through a hard-coded safety filter based on formal verification. This filter knows the laws of physics and the safety limits of the robot. If the transformer suggests moving the arm through a solid wall, the safety layer says, absolutely not, and forces a stop. It's a way of combining the creative, generalizing power of neural networks with the reliable, predictable nature of traditional control theory.

Corn

It's like having a very imaginative child who's allowed to play, but only inside a very sturdy playpen. I like that. It suggests that as we move into twenty twenty-six and twenty twenty-seven, we're going to see robots that are much more capable of handling messy, unpredictable environments—like a house with a dog and a housemate who leaves his shoes in the middle of the hallway—without needing a human to program every single contingency.

Herman

Precisely. And this leads to some really interesting second-order effects. If robots can learn from video, then every time a human uploads a how-to video to the internet, they are inadvertently training the next generation of robots. We're essentially creating a collective physical memory that any embodied A I can tap into. Remember in episode one hundred and seventy-nine when we talked about the ethics of tourism and how capturing data changes the environment? We're seeing a version of that here. The physical world is being indexed and turned into training data at an incredible rate.

Corn

That's a bit mind-bending. The entire world is becoming a textbook for machines. But let's bring it back to the practical side for a second. If someone is listening to this and they're thinking about how this affects their job or their life, what's the takeaway? Is the message that robots are finally ready for prime time because they've adopted the transformer architecture?

Herman

I think the takeaway is that the intelligence gap is closing faster than the hardware gap. We now have the brains—the transformers and the V L A models—to do some incredibly complex things. The bottleneck now is actually the physical stuff: battery life, motor durability, and the cost of the hardware. But from a software perspective, the synthesis of intelligence in robotics has reached a tipping point. We're moving away from robots that can only do one thing in a factory to robots that can learn to do anything in a home.

Corn

It's the generalization of labor. Just like L L Ms generalized the task of writing and coding, embodied A I is going to generalize the task of physical movement. And it's all built on those same pillars: predicting the next token, finding similarities in high-dimensional vector space, and using attention to filter out the noise. It's amazing how much mileage we're getting out of the transformer architecture. It's like the steam engine of the twenty-first century.

Herman

That's a perfect analogy. The steam engine started by pumping water out of mines, but it ended up powering ships and trains and factories. The transformer started by translating French into English, but now it's powering the movements of humanoid robots. And for our listeners, it's worth noting that if you're working in A I or even just interested in it, understanding these fundamental mechanics—how a vector represents a concept—is your superpower. Whether that concept is a word, a pixel, or a robotic wrist rotation, the math is remarkably consistent.

Corn

I think that's a really empowering way to look at it. It can feel like every week there's a new breakthrough that changes everything, but if you understand the core principles, you can see the threads connecting them all. You realize that a robot picking up a sock is actually doing something very similar to a chatbot finishing your sentence. It's all about navigating a world of patterns.

Herman

Exactly. And speaking of patterns, I've noticed a pattern of people listening to our show but forgetting to leave a review! If you're enjoying these deep dives into the weird world of A I prompts, please take a second to leave us a rating or a review on Spotify or your favorite podcast app. It genuinely helps us reach more curious minds and keeps the show growing.

Corn

It really does. And if you have a question like Daniel's—something that's been bugging you or something you've noticed in the news—head over to myweirdprompts.com and send it our way. We love getting these prompts because they push us to look at things from angles we hadn't considered.

Herman

Definitely. This has been a great one. I think I'm going to go see if Daniel needs help with that bookshelf, though. I suspect his proprioception is a little off today.

Corn

Just make sure he doesn't try to use his kitchen robot to do it yet. We're not quite there. Thanks for listening to My Weird Prompts. I'm Corn.

Herman

And I'm Herman Poppleberry. We'll catch you in the next episode.

Corn

Wait, before we go, Herman, one last thing. You mentioned those V L A models. Are we at the point where a robot could actually watch a video of you making your famous sourdough and replicate it exactly? Because that would save me a lot of time on Sunday mornings.

Herman

You know, it's funny you ask. There was a recent research project where robots learned complex kitchen-style manipulation from hours of demonstration videos. The systems could handle and shape dough-like materials, but still struggled with subtle details like achieving consistent tension in the final shaping. It turns out that tactile feedback—the feel of the dough's elasticity—is a very high-dimensional vector that's hard to capture on video.

Corn

So, your job as the family baker is safe for at least another year?

Herman

For now! But I'm keeping an eye on those haptic sensor developments. Once they start tokenizing the feeling of gluten development, I might be in trouble.

Corn

I'll stick to the store-bought stuff until then. Alright, everyone, thanks for joining us. We'll be back next week to dive into another weird prompt. You can find us on Spotify and at our website, myweirdprompts.com, where you can find our full archive and the R S S feed for subscribers.

Herman

Until next time, stay curious and keep those prompts coming!

Corn

See ya.

Herman

Goodbye!