Figure's Humanoid Learns Tasks Without Knowing Its Envelope

by Nick Clark | Published March 28, 2026 | PDF

Figure AI is building a general-purpose humanoid robot that acquires manipulation and locomotion skills through imitation learning, reinforcement learning, and foundation model integration. The approach targets a humanoid that can learn new tasks from demonstration and language instruction rather than requiring explicit programming for each behavior. The ambition is substantial and the engineering is advancing rapidly. But learned skills do not inherently carry capability self-awareness. A policy that learned to make coffee in training does not know whether it can make coffee right now, with the current gripper condition, battery level, and environmental layout. Capability awareness provides this: persistent envelopes that track what learned skills can actually accomplish in current conditions.

What Figure is building

Figure's approach combines a full-size humanoid form factor with learned control policies for locomotion and manipulation. The robot acquires new skills through demonstrations, language-guided learning, and reinforcement learning in simulation. Foundation model integration enables natural language task instruction and visual scene understanding. The goal is a general-purpose worker that can be deployed to new tasks without task-specific programming.

The learning-based approach means the robot's skill repertoire grows over time. New policies are learned, tested, and deployed. The robot accumulates a library of capabilities. What the learning system does not provide is a persistent model of which skills are reliably executable given the robot's current physical state and environmental conditions. A skill learned in one condition may not transfer to different conditions, and the robot has no mechanism to assess this before attempting execution.

The gap between learned skills and capability awareness

A humanoid that has learned fifty manipulation skills possesses fifty policies. It does not know which of those policies are reliable in the current environment. A skill learned with a fresh gripper may fail with a worn one. A skill trained on rigid objects may fail with deformable ones. A locomotion policy trained on flat surfaces may be unreliable on the current slightly uneven floor. Each learned skill has an implicit capability envelope defined by the training distribution, but that envelope is not represented as explicit state that the robot can query.

The general-purpose deployment context makes this gap critical. A humanoid deployed to a new warehouse encounters conditions that differ from training. Lighting, surface texture, object properties, and spatial layout all vary. The robot has no principled way to assess which of its learned skills are reliable in the new environment. It can attempt tasks and observe success or failure. It cannot predict capability before execution.

As the skill library grows, the gap compounds. More skills means more potential capability. But without capability tracking for each skill under current conditions, more skills also means more ways to fail unpredictably.

What capability awareness provides

The capability envelope tracks each learned skill as a multi-dimensional state variable that evolves with conditions. For each skill, the envelope maintains current reliability estimates given the robot's physical state, environmental conditions, and recent execution history. A manipulation skill's envelope narrows as the gripper wears and widens after tool replacement. A locomotion skill's envelope contracts on unfamiliar surfaces and expands as the robot accumulates successful experience.

Temporal forecasting projects how skill reliability will change. Battery depletion affects motor precision, which affects manipulation skill envelopes. Environmental changes over a shift affect locomotion skill reliability. The joint condition of capability, time, and uncertainty enables the robot to accept or decline tasks based on whether its current skill set can reliably accomplish them in the projected conditions.

The structural requirement

Figure AI's vision of a general-purpose humanoid that learns new skills is ambitious and architecturally sound. The structural gap is the capability layer that tells the robot what its learned skills can actually do right now. Capability awareness as a computational primitive transforms a skill-learning humanoid into a self-aware one that tracks the reliability of its skill library under current conditions, forecasts capability changes, and deploys learned skills only when the capability envelope confirms they are executable.

Capability Awareness All 21 steps →

Know what you can do before you try.

Primary Technical Disclosure
○ Capability-, Time-, and Uncertainty-Aware Execution in Autonomous Computational Networks
Secondary Technical
○ Capability as First-Class Computational State○ Capability Envelope for Substrates○ Temporal Executability Forecasting○ Uncertainty as First-Class Propagated Variable○ Capability Envelope Negotiation○ Capability Genealogy Tracking○ Biological Capability Extension○ Network-Level Capability Pressure○ Capability-Permission Distinction○ Capability-Native Computation○ Execution Synthesis and Non-Synthesis○ Agent Behavior Under Constraints○ Predictive Network Planning○ Multi-Agent Contention Resolution○ Capability Robustness Mechanisms○ Capability-Modulated Discovery Traversal○ Capability as Confidence Input○ Embodied Capability Envelopes○ Substrate Resource Negotiation
Applications (General)
○ Robotic Capability Assessment Before Commitment○ Edge Computing Resource Governance Through Capability Envelopes○ Capability Awareness for Surgical Robotics○ Capability Awareness for Agricultural Robotics○ Capability Awareness for Mining Operations○ Capability Awareness for Offshore Energy Operations○ Capability Awareness for Warehouse Logistics Robotics○ Capability Awareness for Construction Robotics
Applications (Specific)
○ Tesla FSD Does Not Know What It Cannot Do○ John Deere's Autonomous Tractors Cannot Assess Their Own Limits○ KUKA Robots Execute Without Knowing Their Envelope○ FANUC Robots Have No Adaptive Capability Envelope○ Universal Robots Cobots Execute Without Knowing Their Limits○ ABB Robots Perform Without Self-Assessing Capability○ Yaskawa Motoman Robots Move Without Tracking Capability Drift○ Doosan Cobots Collaborate Without Capability Self-Knowledge○ Agility Robotics' Digit Walks Without Knowing What It Can Carry● Figure's Humanoid Learns Tasks Without Knowing Its Envelope
Capability Awareness overview →
AQ
deterministic
autonomy

Last updated: 2026-03-03

-
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie