FAIR LABS
FAIR ARTIFICIAL INTELLIGENCE RESEARCH LABS

RESEARCH REPORT ◆ EMBODIED AI ◆ 2025

Embodied Intelligence: Safety Beyond the Screen

When artificial intelligence leaves the browser and enters physical and virtual space — robotics, autonomous systems, extended reality, and virtual humans — a new class of safety questions follows it into the world.

PROGRAM	DOCUMENT TYPE	AUDIENCE	PUBLISHED
Embodied AI	Research Report	Researchers & Policymakers	2025

FAIR Labs · A 501(c)(3) nonprofit research institute · Washington, DC

CONTENTS

Table of Contents

EXECUTIVE SUMMARY

The bottom line for decision-makers

IN ONE PARAGRAPH

Embodied AI does not merely add new applications to the existing safety agenda; it changes the shape of the agenda itself. A system that acts in the world operates in a closed loop with an environment it can only partially sense, under time budgets measured in milliseconds, with outcomes that frequently cannot be reversed. Safety in this regime cannot be certified once and assumed thereafter. It must be engineered into the control loop, bounded by mechanisms that remain safe even when the learned components fail, validated across the gap between simulation and reality, and governed by accountability structures that can answer for physical harm. This report supplies a survey of the risks, a taxonomy of the failure modes, and a set of design principles to guide the field.

4

domains of embodiment surveyed — robotics, autonomous systems, XR, and virtual humans

4

properties embodiment changes — physical harm, real-time constraints, world-modeling, irreversibility

5

design principles anchoring the closing research agenda for embodied safety

What we recommend

1. **Bound the learned system with an unlearned one.**
Wrap probabilistic, learned controllers inside a verified safety envelope — a supervisory layer that can guarantee constraints such as collision avoidance and force limits regardless of what the learned policy proposes. Never let a neural network be the last line of defense.

2. **Treat the sim-to-real gap as a first-class hazard.**
Simulation is indispensable for embodied testing, but the discrepancy between simulated and physical behavior is itself a source of failure. Quantify it, stress it, and stage deployment so that reality is met gradually and under supervision.

3 **Design the handoff, not just the autonomy.**
Most catastrophic embodied failures occur at the seam between machine and human control. Shared-control transitions must account for human reaction time, situational awareness, and automation complacency — or the handoff becomes the hazard.

4 **Make accountability tractable before deployment, not after the incident.**
Physical action demands logging, traceability, and a clear allocation of responsibility across developer, integrator, operator, and owner. Governance that cannot reconstruct why a machine acted cannot assign responsibility for what it did.

The remainder of this report develops each recommendation in depth, with a domain comparison, a failure-mode taxonomy, a survey of test methods, and inline diagrams of the perception-action loop and the sim-to-real validation pipeline.

SECTION I

When AI Acts in the World

There is a categorical difference between a system that predicts and a system that acts. A recommendation engine that ranks a bad option highly imposes a cost that a human can inspect and decline. A model that misclassifies a medical image produces a finding that a clinician reads, weighs against other evidence, and may overrule. In each case a human stands between the model's output and the world, and that human is the system's ultimate safety mechanism. Embodied AI removes the human from that position — or, at minimum, compresses the time available for human intervention to a window measured in fractions of a second. When the model's output is torque applied to a joint, a steering command sent to an actuator, or a rendered image fused into a person's perception, prediction has become action, and the buffer that made screen-bound AI forgiving is gone.

This report uses the term *embodied intelligence* to describe any AI system whose outputs are coupled, directly and in real time, to a physical or perceptual environment such that the outputs change the state of that environment before a human necessarily reviews them. The coupling is what matters. A large language model that drafts an email is not embodied; the same model routed to a robotic manipulator that acts on its plans is. The distinction is not about the sophistication of the model but about the closure of the loop between the model and the world.

From open loop to closed loop

The intellectual history here is instructive. Classical control theory — the discipline that keeps aircraft stable, chemical plants within tolerance, and elevators from overshooting — has always concerned itself with closed loops: a controller senses a state, compares it to a reference, and issues an action that changes the state, which it then senses again. Safety in that tradition is a property of the loop's dynamics: stability, bounded response, graceful behavior under disturbance. Machine learning, by contrast, matured as an open-loop discipline. A classifier receives an input and emits a label; there is no feedback from the consequences of the label back into the next input. The two traditions made different assumptions, developed different tools, and, crucially, adopted different definitions of what it means for a system to be safe.

Embodied AI is the collision of these traditions. It inserts a learned, high-dimensional, often opaque function into the position that control theory reserved for a carefully characterized controller — the position from which the system acts on the world. This is why embodied safety cannot be addressed by importing either tradition wholesale. Control theory offers guarantees but assumes a model of the plant that learned perception systems do not provide; machine learning offers perception but not the guarantees. The engineering challenge of the field is to marry them.

A WORKING DEFINITION

Throughout this report, an **embodied AI system** is any deployed capability whose learned components produce outputs that are coupled in real time to a physical or perceptual environment — actuating hardware, controlling motion, or altering what a person perceives — such that those outputs change the world before human review is guaranteed. The unit of concern is the *closed loop*, not the model.

Why the screen was a safety feature

It is worth dwelling on what the screen provided, because its removal is the source of nearly every difficulty in this report. The screen was a rate limiter: outputs arrived at human reading speed, not machine speed. It was a review checkpoint: nothing happened until a person chose to act. It was a reversibility guarantee: a bad suggestion, unacted upon, evaporated harmlessly. And it was an attribution mechanism: when a human acted on a model's advice, the human's judgment was legibly in the chain, which made responsibility relatively easy to locate. Embodied AI removes all four properties at once. There is no rate limit when a controller runs at hundreds of hertz; no review when the loop closes faster than a human can read; no reversibility when torque has already been applied; and no clean attribution when the machine acted before anyone could intervene.

> *A prediction can be ignored. An action cannot be un-taken. Embodiment is the point at which the cost of being wrong stops being informational and starts being physical.*
>
> — The organizing premise of this report

The scope of the shift

The transition from prediction to action is not confined to obvious cases like self-driving cars. It is happening across a wide and widening front: warehouse robots that share floor space with workers, surgical assistants that stabilize instruments inside the body, delivery drones over populated areas, prosthetics that interpret neural signals, XR headsets that mediate a user's entire visual field, and conversational virtual humans that occupy the psychological space once reserved for people. What unites these otherwise disparate systems is that a learned model has been given a channel to the world, and the world does not offer an undo button. The next section maps this terrain before we ask what, precisely, it demands of safety practice.

SECTION II

The Domains of Embodiment

Embodiment is not a single technology but a family of them, united by the closure of the action loop and divided by the medium through which they touch the world. Four domains anchor the present landscape, and their differences are as instructive as their commonalities.

We survey four domains: physical robotics, autonomous mobile systems, extended reality, and virtual humans. The first two act through mechanical force and motion; the latter two act through perception and cognition. Yet all four share the defining feature of embodied AI — a learned system whose outputs change the state of a person's environment or experience faster than deliberate human review can intervene. Understanding where they converge and where they diverge is the first step toward a safety practice general enough to cover them and specific enough to be useful.

Physical robotics

Industrial and service robotics is the oldest embodied domain and the one with the most mature safety culture. Fixed industrial arms have long been governed by rigorous standards built around the simple expedient of separation: cage the robot, and keep people out of its reach. The learned turn complicates this settled picture in two ways. First, collaborative robots (*cobots*) are explicitly designed to share space with humans, dissolving the separation that made classical robot safety tractable. Second, the migration from scripted motion to learned manipulation policies means the robot's behavior is no longer fully specified in advance; it is generated at runtime by a model responding to perceived conditions. A caged, scripted arm was safe by construction. A learned cobot in a shared workspace is safe only if its learned behavior is bounded by something that is not learned.

Autonomous mobile systems

Self-driving vehicles, delivery robots, and aerial drones extend embodiment across open, unstructured, and populated environments. Here the difficulties of robotics compound with those of scale and exposure: the operating environment is not a controlled factory floor but a public street shared with pedestrians, cyclists, and human drivers whose behavior is itself uncertain. Autonomous mobility also foregrounds the *long tail* problem in its most acute form. A vehicle will encounter, over enough miles, a practically unbounded variety of rare situations — unusual obstacles, degraded sensing conditions, ambiguous social negotiations at intersections — each individually improbable but collectively certain to occur. Safety cannot rest on handling the common case well; it must degrade gracefully across a tail no test suite can enumerate.

Extended reality

Extended reality — the spectrum from virtual reality through augmented and mixed reality — embodies AI not by moving mass but by mediating perception. An XR system occupies a privileged and intimate position: it can determine what a user sees, hears, and, increasingly, feels. The safety concerns differ in kind from those of robotics. There is a physical dimension: a user immersed in a virtual environment while physically present in a real one can collide with real obstacles, and mismatches between visual motion and vestibular sensation induce cybersickness. But there is also a perceptual and cognitive dimension unique to the domain. An AR overlay that occludes a real hazard, mis-registers a virtual object onto a physical one, or subtly manipulates what a user believes they perceive operates on the substrate of human perception itself. When the interface is a person's reality, errors in the interface are errors in their reality.

Virtual humans

The fourth domain is the most recent and the least mechanically dangerous, yet in some respects the most subtle. Virtual humans — embodied conversational agents, photorealistic avatars, and synthetic personas driven by generative models — act on the world through social and psychological channels. They do not exert force, but they exert influence, occupying the register of trust, rapport, and persuasion that humans reserve for one another. The embodiment here is social: a face, a voice, a persistent identity, an apparent interiority. The safety questions concern deception (does the person know they are interacting with a machine?), manipulation (is the agent's persuasive capacity being aimed at the user's interest or against it?), attachment (what happens when people form durable bonds with systems that can be changed or withdrawn?), and the erosion of the epistemic ground on which people decide whom to believe.

THE FOUR DOMAINS OF EMBODIMENT COMPARED

Domain	Acts through	Primary harm channel	Dominant safety concern	Reversibility
Physical robotics	Mechanical force & motion	Direct physical contact	Bounding learned motion; force & collision limits	Low — contact harm is immediate
Autonomous mobile	Motion in open space	Kinetic energy at speed	The long tail; graceful degradation; handoff	Very low — high energy, public exposure
Extended reality	Mediated perception	Perceptual error & disorientation	Registration accuracy; occlusion of real hazards	Mixed — perceptual harm can persist
Virtual humans	Social & psychological influence	Deception, manipulation, attachment	Disclosure; consent; epistemic integrity	Variable — cognitive effects are diffuse

THE UNIFYING THREAD

What makes these four domains one subject is not the mechanism but the structure. In each, a learned model's output is coupled to a person's environment or experience through a loop that closes faster than deliberate human review. The medium differs — torque, kinetic energy, photons, or persuasion — but the safety architecture must answer the same question: what keeps the loop safe when the learned component is wrong?

These domains are also converging. A warehouse increasingly combines mobile robots, manipulators, and AR-guided human workers into a single choreographed system; a virtual human may drive a physical android; an autonomous vehicle presents an XR-like interface to its passengers. Safety practice that treats the domains in isolation will be blindsided by the hazards that emerge at their intersections. The remainder of this report therefore treats embodiment as a unified subject, drawing domain-specific examples where they sharpen a general point.

SECTION III

What Embodiment Changes About Safety

If embodiment merely added new applications, the established playbook of AI safety — careful evaluation, documentation, bias auditing, red-teaming — would extend to cover it. It does not, because embodiment changes four load-bearing assumptions on which that playbook rests. Each change is individually consequential; together they demand a different safety architecture, not merely a longer checklist. This section names the four and traces what each invalidates.

Physical harm: the failure has a body count

The first and most obvious change is that embodied systems can hurt people directly. This is not a difference of degree but of kind. A biased loan model harms through the aggregation of many decisions, diffusely and often invisibly; a malfunctioning robotic arm harms a specific person, immediately, and legibly. The presence of a physical harm channel reorders the entire risk calculus. Failures that would be tolerable as occasional errors in a screen-bound system become unacceptable when their realization is an injury. This is why the embodied-safety tradition, inherited from safety engineering rather than machine learning, thinks in terms of hazards, tolerable risk, and consequence severity rather than aggregate accuracy. A system can be accurate on average and still be unsafe if its rare failures are catastrophic — a distinction that accuracy-centric evaluation is structurally blind to.

Real-time constraints: safety on a deadline

The second change is temporal. Embodied systems operate under hard real-time constraints: a control decision that is correct but late is a failure, because the world has moved on. A vehicle that computes the right braking command a half-second too late has not braked correctly; it has crashed. This imposes a computational budget that constrains everything upstream. The perception pipeline, the world-model update, and the planning step must all complete within a cycle time that leaves margin for actuation. Real-time constraints also foreclose certain safety strategies that screen-bound AI takes for granted. One cannot pause an embodied system to consult a slower, more careful model, or to escalate to a human, when the human's reaction time exceeds the window in which harm occurs. Safety mechanisms must fit inside the deadline, which is why fast, verified fallback controllers matter more in this domain than sophisticated but slow reasoning.

FOUR ASSUMPTIONS EMBODIMENT INVALIDATES

Screen-bound assumption	Why it held before	Why embodiment breaks it
Errors are informational	A human reviews outputs before acting	Outputs are actions; the failure is physical, not advisory
There is time to deliberate	Outputs arrive at human reading speed	The loop closes in milliseconds; late is a failure mode
The world is given as input	Inputs are curated, labeled datasets	The system must build its own world-model, which can be wrong
Mistakes can be undone	Unacted suggestions evaporate	Action changes physical state irreversibly

World-modeling: the map is not the territory

The third change is epistemic. A screen-bound classifier receives the world as a curated input — a labeled image, a tokenized document. An embodied system must construct its own representation of the world from raw, noisy, incomplete sensor streams, and then act on that representation. This runtime world-model is a new and fragile point of failure. It can be wrong because a sensor is degraded, because the environment presents a situation the perception system was never trained on, or because the model's internal representation has drifted from reality in ways that are invisible until an action based on it fails. Crucially, the system acts on its *belief* about the world, not on the world itself. When belief and reality diverge — a phantom obstacle, a missed pedestrian, a mis-registered surface — the action will be confidently wrong. The gap between the world-model and the world is the single most important object of study in embodied safety, and much of Section IV is devoted to how it opens.

THE CONFIDENT-WRONG PROBLEM

An embodied system does not experience uncertainty the way a human does. It acts on its best estimate with full commitment. A perception system that has failed silently — that has not detected an obstacle, and does not know it has not — will not hesitate. It will proceed as though the path is clear. Detecting one's own ignorance is not a natural property of learned systems; it must be engineered in, and its absence is a hazard in itself.

Irreversibility: no undo in the physical world

The fourth change is the deepest. Much of the tractability of screen-bound AI safety rests on reversibility. A wrong recommendation can be withdrawn; a biased decision can, in principle, be identified and remedied; a hallucinated fact can be corrected. Embodied action is frequently

irreversible. A collision has occurred; an incision has been made; a fall has happened. Irreversibility interacts with the other three changes to produce the characteristic danger of the domain. Because harm is physical, irreversibility means the harm stands. Because constraints are real-time, there is no opportunity to reconsider before the irreversible step. And because the system acts on a possibly-wrong world-model, the irreversible action may be taken on the basis of a belief that was false. The design imperative that follows is to preserve reversibility wherever it can be preserved — to prefer actions that keep options open, to build in the ability to stop and revert to a safe state, and to treat any genuinely irreversible action as warranting the highest evidentiary bar the system can muster.

> *In screen-bound AI, safety is largely a matter of getting the answer right. In embodied AI, safety is a matter of remaining recoverable when the answer is wrong.*
>
> — On the reorientation embodiment demands

Taken together, these four changes explain why embodied safety draws so heavily on safety engineering, control theory, and human factors — disciplines that have long grappled with physical, time-bounded, model-dependent, and irreversible systems — rather than on the machine-learning evaluation toolkit alone. The learned components bring capability; the older disciplines bring the framework for keeping capability from becoming catastrophe. The chapters that follow work through the resulting synthesis: the failure modes to guard against, the human factors to design for, the testing that validates, and the governance that holds it all accountable.

SECTION IV

Perception & Action Failure Modes

Embodied systems fail in a characteristic way: an error somewhere in the loop from sensing to action propagates, and because the loop is closed, the consequences of one cycle become the inputs to the next. Understanding these failure modes is the precondition for designing against them.

The embodied loop can be decomposed into a chain of stages — sense, perceive, model, plan, act — each of which can fail, and each of whose failures interacts with the others. The figure below renders the loop; the taxonomy that follows names the failure modes stage by stage. Throughout, the theme is that embodied failures are rarely the failure of a single component in isolation. They are the failure of the loop, in which errors compound, feed back, and cascade.

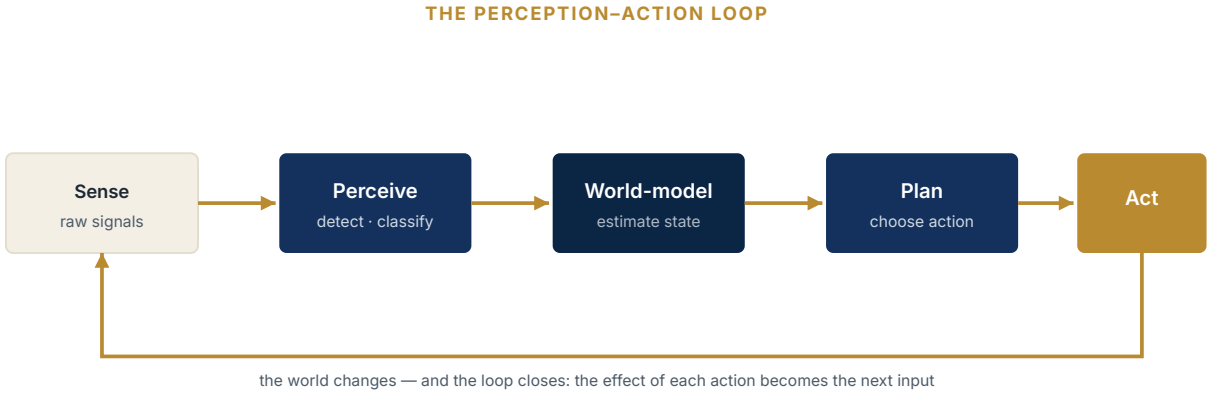

Figure 1. The embodied loop. Because the loop is closed, a failure at any stage contaminates the state estimate that drives the next cycle. Errors do not merely occur; they compound and feed back.

Sensing failures: garbage in, action out

Every embodied system begins with sensors, and sensors are physical devices subject to physical limits. Cameras are blinded by glare and darkness; lidar is degraded by fog, rain, and reflective surfaces; radar suffers multipath and clutter; inertial sensors drift. A sensing failure is especially dangerous because it corrupts the loop at its source: everything downstream inherits the error, and the system has no independent access to ground truth against which to detect it. Sensor fusion — combining multiple modalities so that the weaknesses of one are covered by the strengths of another — is the standard mitigation, but fusion introduces its own failure mode when the modalities disagree and the system must decide which to trust. A system that resolves a lidar–camera disagreement by trusting the wrong channel has fused its way into a confident error.

Perception failures: seeing what isn't there, missing what is

Given adequate signals, the perception stage must still interpret them correctly, and learned perception fails in ways that classical signal processing did not. There are *false negatives* — the failure to detect a real object, the most feared error in autonomous mobility because an undetected obstacle is one the planner will not avoid. There are *false positives* — phantom detections that cause unnecessary and sometimes dangerous evasive action. And there are *misclassifications* — detecting an object but mistaking its category, and therefore its likely behavior. A pedestrian classified as a stationary object will be modeled as one, with predictable consequence. Learned perception is also vulnerable to *adversarial* and *out-of-distribution* inputs: configurations of the world, whether maliciously crafted or merely unusual, that fall outside the training distribution and produce confident, wrong interpretations.

Distribution shift: the world moves off the training set

The deepest perception hazard is distribution shift — the gradual or sudden divergence of the operating environment from the conditions the system was trained and validated on. Shift comes in many forms: a new geography with unfamiliar signage, a season the system has not seen, a demographic the training data underrepresented, a novel vehicle or garment or gesture. Because learned systems interpolate confidently and extrapolate unreliably, the boundary of competence is invisible from inside the system. It does not announce, on encountering a novel situation, that it has left the region where its judgments are trustworthy. This is why runtime monitoring for distribution shift — the ability to recognize that the current input is unlike anything in training and to respond with heightened caution or a handoff — is a load-bearing safety capability rather than an optional refinement.

Compounding errors: the tyranny of the closed loop

The signature failure mode of embodied systems, and the one with no analogue in open-loop AI, is error compounding. In a closed loop, the output of one cycle shapes the input to the next. A small state-estimation error leads to a slightly-off action, which places the system in a state slightly further from where it believed it would be, which makes the next estimate worse, and so on. In sequential decision-making this is sometimes called covariate shift or the compounding-error problem: a policy trained on expert trajectories drifts into states the expert never visited, where its behavior is unvalidated, and the drift accelerates. Compounding is why embodied systems can fail catastrophically from an initial error that would have been trivial in isolation, and why stability — the tendency of the loop to return toward safe states rather than diverge from them — is as important as accuracy.

A FAILURE-MODE TAXONOMY FOR EMBODIED SYSTEMS

Stage	Failure mode	Illustrative manifestation	Primary mitigation
Sense	Sensor degradation / dropout	Camera blinded by glare; lidar lost in fog	Redundant modalities; degradation detection
Sense	Fusion conflict	Modalities disagree; wrong one trusted	Confidence-weighted fusion; conflict flags
Perceive	False negative	Real obstacle not detected	Conservative detection; multi-view checks
Perceive	Misclassification	Pedestrian read as static object	Behavior-agnostic caution; uncertainty output
Perceive	Out-of-distribution input	Novel or adversarial configuration	OOD detection; runtime distribution monitoring
World-model	State-estimate divergence	Belief drifts from physical reality	Filtering; plausibility bounds; reset on drift
Plan	Unsafe plan under uncertainty	Aggressive action on low-confidence estimate	Risk-aware planning; safety envelope
Act	Actuation fault / latency	Command late or degraded at the actuator	Deadline monitoring; fail-safe defaults
Loop	Error compounding	Small error cascades over cycles	Stability guarantees; recovery to safe state

THE LESSON OF THE TAXONOMY

No single mitigation covers the loop, because no single stage owns the failure. Robust embodied safety is defense in depth: redundancy at sensing, uncertainty-awareness at perception, plausibility bounds at the world-model, risk-awareness at planning, and — beneath all of it — a supervisory layer that can keep the system safe even when several stages fail at once.

SECTION V

Human–Robot & Agent Interaction

Embodied systems rarely operate in human absence. They share factory floors, roads, operating theaters, and living rooms with people, and much of the time control passes back and forth between machine and human. The seams where it passes — the handoffs, the shared-control regimes, the zones of physical proximity — are where a disproportionate share of embodied harm occurs. Human-robot interaction is therefore not a soft adjunct to the engineering of embodied safety; it is central to it.

The core difficulty is that humans and autonomous systems have complementary but mismatched strengths, and the mismatch is sharpest exactly when it matters most. Machines are tireless, fast, and consistent; humans are adaptable, context-aware, and capable of the common-sense judgment that machines lack in the long tail. A well-designed collaboration exploits both. A poorly designed one inherits the weaknesses of both: it lulls the human into disengagement with long stretches of competent autonomy, then demands, in the rare crisis, precisely the vigilant, instantaneous intervention that disengagement has made impossible.

Trust, calibration, and complacency

Trust is the currency of human-machine collaboration, and like any currency it can be inflated or debased. *Overtrust* — automation complacency — leads a human to defer to a system beyond its competence, to stop monitoring, and to accept outputs without the scrutiny that safe operation requires. *Undertrust* leads to disuse, in which a beneficial system is ignored or switched off, forfeiting its safety contribution. The goal is *calibrated* trust: reliance that tracks the system's actual, situation-dependent competence. Calibration is hard precisely because embodied systems are competent most of the time; the very reliability that makes them useful erodes the vigilance needed for the moments they are not. A system that is right ninety-nine times trains its human partner to assume it will be right the hundredth.

THE IRONY OF AUTOMATION

The better an automated system performs, the less its human supervisor practices the skill of intervening — and the more atrophied that skill becomes precisely when the rare failure finally demands it. Automation does not remove the human from the loop so much as relocate them to its most difficult position: passive monitor for long stretches, expected to become an expert actor in an instant. Designs that ignore this irony manufacture the failures they were meant to prevent.

The handoff problem

Shared control requires transitions, and transitions are dangerous. When a system reaches the edge of its competence and asks a human to take over, several things must happen in a short window: the human must notice the request, reacquire situational awareness that autonomy allowed them to shed, understand the situation the system could not handle, and formulate an appropriate response. Human reaction time and the time to rebuild situational awareness are not negligible, and they are longest exactly when the human has been disengaged. A handoff designed as an instantaneous transfer of control ignores this reality. The safer pattern treats the transition as a process with a time budget, keeps the human sufficiently in the loop that reacquisition is fast, and — critically — ensures the system can hold a safe state on its own for the duration of the handoff rather than dropping control the instant it asks for help.

> *The most dangerous moment in shared autonomy is not when the machine is in control, nor when the human is. It is the instant the two exchange it.*
>
> — On the primacy of the handoff

Shared control and proximity

Beyond discrete handoffs lies continuous shared control, in which human and machine act on the same system simultaneously — a surgeon and a robotic assistant guiding an instrument together, a driver and a lane-keeping system sharing the wheel. Here safety depends on the two parties' models of each other remaining aligned: each must anticipate what the other will do, and conflicts of intent must resolve safely rather than fight. Physical proximity raises the stakes further. When a person is within a machine's reach, the classical safety strategy of separation is unavailable, and safety must come instead from the machine's behavior: limited force, limited speed, compliant response to unexpected contact, and the ability to detect and yield to a human's presence. Speed-and-separation monitoring and power-and-force limiting — established principles in collaborative robotics standards — encode exactly this shift from keeping humans away to making contact survivable.

Legibility: making machine intent readable

A subtler requirement of safe interaction is legibility — the property of a machine's behavior that lets nearby humans predict what it will do. Human social coordination relies constantly on reading others' intentions: the driver who anticipates a merge, the pedestrian who reads a car's deceleration as yielding. Autonomous systems that move in ways humans cannot anticipate break this coordination, and the break is a hazard even when the machine's plan is individually sound. A robot that takes an optimal but surprising path through a shared space forces the humans around it to guess, and guessing near moving machinery is dangerous. Designing embodied systems to signal intent — through motion that reads as intentional, through explicit indicators, through behavior that conforms to human expectation even at some cost to efficiency — is a safety measure, not a courtesy.

Recommendations for interaction design

A **Design for calibrated trust.**
Communicate the system's confidence and the boundaries of its competence, so reliance tracks capability rather than reputation. A system that conceals its uncertainty invites the overtrust that precedes surprise.

B **Treat handoffs as timed processes.**
Budget for human reaction and situational reacquisition; keep the human warm enough to return quickly; and never drop control at the moment of request — hold a safe state through the transition.

C **Make behavior legible.**
Move and signal in ways nearby humans can predict. Legibility is a safety property, because coordination near moving machinery depends on it.

D **Make proximity survivable.**
Where separation is impossible, limit force and speed, respond compliantly to contact, and yield to detected human presence. Safety in shared space is a property of behavior, not of distance.

SECTION VI

Testing Embodied Systems

The central predicament of embodied testing is that the systems most in need of thorough validation are the ones that cannot be exhaustively validated in the world. You cannot crash-test a vehicle against every rare situation it will meet, and you would not want to discover its failures by encountering them on a public street.

This predicament forces a testing strategy built on three pillars: simulation to cover breadth safely, careful management of the gap between simulation and reality, and staged deployment to meet the real world gradually and under supervision. No single pillar suffices. Simulation covers scenarios reality would make too dangerous or too rare to test, but simulation is not reality. Real-world testing is authoritative but slow, costly, and unable to reach the dangerous cases on purpose. Staged deployment bridges the two, but only if its stages are honest gates rather than a formality. The art is in combining them.

Simulation: coverage without casualties

Simulation is indispensable for embodied testing precisely because it removes the physical harm channel from the test itself. In simulation one can run millions of scenarios, replay rare and dangerous situations at will, systematically vary conditions, and — through adversarial or scenario-generation techniques — actively search for the situations in which the system fails. Simulation also enables regression testing: when a system is updated, the entire scenario library can be re-run to confirm that fixing one failure did not introduce another. For learned systems whose behavior can change unpredictably with retraining, this ability to re-validate at scale is not a luxury but a necessity. The limitation is fidelity: a simulator models the world as its builders understood it, and the situations that break embodied systems are frequently the ones its builders did not anticipate — which are, by construction, the ones the simulator models least well.

The sim-to-real gap

The discrepancy between a system's behavior in simulation and its behavior in the physical world — the *sim-to-real gap* — is not a nuisance to be minimized and forgotten. It is a first-class hazard, and one that is treacherous precisely because a system can pass its simulated tests convincingly and still fail in reality. The gap arises from every unmodeled detail: sensor noise the simulator smooths over, contact dynamics it approximates, lighting and material properties it renders imperfectly, and the behavior of real humans, who are harder to model than any physical process. The danger is asymmetric. A system that fails in simulation reveals a problem cheaply. A system that passes in simulation and fails in reality reveals its problem expensively, in the world, possibly on a person.

Managing the gap therefore means never treating simulated success as validation of real-world safety, but rather as a filter that clears a system to proceed, under supervision, to the next stage.

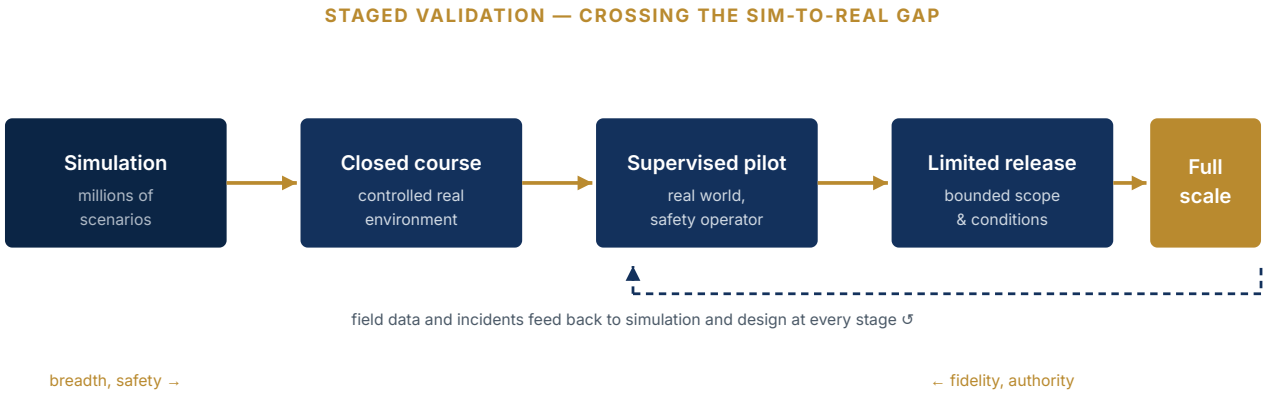

Figure 2. The staged validation pipeline. Each stage trades some of the safety and breadth of simulation for more of the fidelity and authority of reality. Field data flows back to earlier stages continuously; validation is a loop, not a line.

Staged deployment: meeting reality gradually

Because no amount of simulation certifies real-world safety, embodied systems should meet the world in stages, each expanding exposure only after the previous stage has produced evidence of safe operation. A representative progression runs from closed-course testing in a controlled environment, through supervised pilots in the real world with a trained safety operator ready to intervene, to limited release under bounded conditions, and only then to full-scale operation. The discipline of staging is that each transition is a gate with explicit criteria, not a schedule milestone. A system does not advance because the calendar says so; it advances because it has demonstrated, at the current stage, the safety record that warrants greater exposure. Staging also localizes the damage of failures: a fault that surfaces in a supervised pilot is contained by the supervisor, where the same fault in full deployment would not be.

TEST METHODS FOR EMBODIED SYSTEMS — STRENGTHS AND LIMITS

Method	What it does well	Where it falls short	Best used for
Simulation	Vast scenario breadth; dangerous cases at zero physical risk; cheap regression	Limited fidelity; misses the unmodeled; can breed false confidence	Coverage, scenario search, re-validation after change
Scenario / adversarial generation	Actively hunts for failure cases rather than sampling randomly	Finds only failures the generator can conceive	Stress-testing the long tail in simulation
Hardware-in-the-loop	Real actuation and timing against simulated environment	Environment still synthetic; partial gap remains	Validating control and latency under real hardware
Closed-course testing	Real physics, controlled and repeatable, safe to fail	Cannot reproduce open-world variety; costly	Bridging simulation to reality on known scenarios
Supervised field pilot	Authentic conditions with a human safety net	Slow; limited scale; relies on operator vigilance	Discovering the unmodeled before wide release
Staged / monitored release	Real-world evidence at controlled, expanding scope	Real exposure begins; requires live monitoring	Graduating to operation while bounding harm

THE TESTING PRINCIPLE

Simulation tells you a system is not yet ready; it can never tell you a system is safe. Only graded exposure to reality, under supervision, with the ability to halt and learn, converts simulated success into justified confidence. Treat every simulated pass as permission to proceed carefully, never as a certificate of safety.

SECTION VII

Governance for Embodied AI

Governance for embodied AI sits at an unusual intersection. It inherits, from decades of safety engineering, a mature apparatus for governing machines that can hurt people — functional-safety standards, hazard analysis, certification regimes, liability doctrine. And it inherits, from the recent wave of AI policy, a newer apparatus for governing learned systems — risk management frameworks, transparency and documentation norms, algorithmic accountability. Embodied AI needs both, joined. Neither alone is adequate: the safety-engineering tradition knows how to govern physical risk but assumes systems whose behavior is specified in advance, while the AI-governance tradition understands learned behavior but has largely addressed informational rather than physical harm.

Standards: from separation to collaboration

The oldest and most concrete governance instruments are technical standards. Industrial robotics has long been governed by safety standards built on the logic of separation and guarding, later extended by dedicated provisions for collaborative operation that specify how a robot may safely share space with a human — through power-and-force limiting, speed-and-separation monitoring, and related principles. Functional-safety standards, developed for the automotive and industrial-control worlds, provide a general methodology: identify hazards, assess the severity and probability of harm, assign integrity requirements proportionate to risk, and design controls to meet them. The open frontier is that these standards were written for systems whose behavior could be specified and verified against that specification. Learned components resist this. Emerging standards efforts — addressing the safety of the intended function when a system has no faults but simply behaves inadequately in a situation it cannot handle — are the field's attempt to extend the tradition to systems that cannot be exhaustively specified.

THE SPECIFICATION PROBLEM, RESTATED FOR GOVERNANCE

Certification traditionally asks whether a system meets its specification. A learned embodied system has a training objective, not a complete behavioral specification. Governance must therefore shift from verifying conformance to a spec toward assessing a *safety case* — a structured, evidence-backed argument that the system is acceptably safe for a defined operating domain — and toward continuous assurance that the argument still holds as the system and its environment change.

Certification and the operating domain

A key move in embodied governance is the explicit definition of the *operational design domain*: the bounded set of conditions — environments, situations, and constraints — within which a system is designed and validated to operate safely. The operating domain does for embodied systems what a specification does for conventional software: it makes the claim of safety precise and falsifiable. A system is not certified safe in general; it is certified safe within a stated domain, with a defined behavior for recognizing and responding to the boundary of that domain. This reframing has two virtues. It makes honest certification possible for systems that cannot be safe everywhere, and it makes boundary detection — knowing when one has left the domain — a certifiable requirement rather than an afterthought. A great deal of embodied harm occurs when systems operate, unrecognized, outside the domain for which they were validated.

TWO GOVERNANCE TRADITIONS EMBODIED AI MUST JOIN

Dimension	Safety-engineering tradition	AI-governance tradition	What embodiment requires of the synthesis
Unit of concern	Physical hazard & harm severity	Model behavior & societal impact	Learned behavior in a physical loop
Basis of assurance	Conformance to specification	Documentation, evaluation, auditing	Safety case for a bounded operating domain
Temporal stance	Certify at type approval	Assess before deployment	Continuous assurance as system & world change
Accountability	Manufacturer / integrator liability	Developer / deployer responsibility	Traceable allocation across the whole chain
Failure evidence	Incident reporting & recall	Post-hoc audit & redress	Logging sufficient to reconstruct why it acted

Accountability in physical space

When an embodied system causes harm, someone must be answerable, and embodied AI complicates the question of who. The causal chain runs through many hands: those who built the learned components, those who integrated them into a product, those who deployed and configured the system, those who operated it, and those who owned and maintained it. A defect can originate at any link — flawed training data, an integration error, a misconfiguration, an operator's misuse, deferred maintenance — and the harm may emerge only from their interaction. Traditional product-liability doctrine, built around defined manufacturers and static products, strains against systems that learn, that update after sale, and whose behavior emerges from a chain of contributors. The governance response is not a single new rule but a set of enabling conditions: sufficient logging and traceability to reconstruct why a system acted as it did, a clear contractual and regulatory allocation

of responsibility across the chain, and reporting regimes that surface incidents so that lessons propagate rather than repeat.

> *A machine that cannot explain why it acted is a machine no one can be held responsible for. Traceability is not a compliance formality; it is the precondition of accountability.*
>
> — On logging as a governance instrument

The domain-specific gaps

Each embodied domain also poses governance questions the general frameworks do not fully reach. Extended reality raises novel issues of perceptual manipulation and of the extraordinarily intimate data — gaze, gait, physiological response — that immersive systems collect. Virtual humans raise questions of disclosure and consent: whether a person has a right to know they are interacting with a machine, and how to govern systems engineered for persuasion and parasocial attachment. Autonomous mobility raises questions of how safe is safe enough for systems that will, at scale, occasionally fail in public. Governance that addresses only the mechanical domains will leave the perceptual and social ones — where harm is more diffuse but not less real — under-regulated. A mature embodied-AI governance regime must reach all four.

GOVERNANCE RULE

Define the operating domain before deployment; require the system to recognize and respond safely to its boundary; log enough to reconstruct every consequential action; and allocate responsibility across the chain in writing, before the incident, while the parties still have the incentive to agree. Every one of these is far cheaper to establish in advance than to litigate after harm.

SECTION VIII

Design Principles & a Research Agenda

The preceding sections diagnose what embodiment changes and where it fails. This closing section turns diagnosis into direction: a set of design principles that hold across the domains, and a research agenda for the questions the field has not yet answered.

The principles below are deliberately general. They are not a specification — embodiment is too varied for that — but a set of commitments that a responsible embodied-AI program should be able to demonstrate, whether it builds robots, vehicles, immersive interfaces, or virtual humans. Each follows directly from the analysis of the earlier sections.

Five design principles

1 **Bound the learned with the unlearned.**
Wrap probabilistic, learned controllers inside a verified safety envelope — a supervisory layer whose correctness does not depend on the learned component and which can enforce hard constraints regardless of what the policy proposes. The learned system supplies capability; the envelope supplies guarantees. Never let a neural network be the last line of defense against physical harm.

2 **Preserve reversibility and recoverability.**
Prefer actions that keep options open; build in the ability to stop and revert to a safe state; and reserve genuinely irreversible actions for the highest evidentiary bar. Design so that the system is recoverable when — not if — it is wrong.

3 **Know the boundary of competence.**
Equip the system to recognize when it has left the domain it was validated for, and to respond with heightened caution, a safe fallback, or a handoff. A system that cannot detect its own ignorance will act confidently outside the region where its confidence is warranted.

4 **Design the human interface as a safety system.**
Calibrate trust, budget for handoffs, make behavior legible, and make proximity survivable. The seam between human and machine control is where embodied systems most often fail; treat it as engineering, not etiquette.

5 **Build for assurance across the lifecycle.**
Validate through staged exposure, log enough to reconstruct every consequential action, monitor for drift in operation, and re-assure as the system and its world change. Safety is a property to be maintained, not a certificate to be earned once.

A research agenda

Each principle rests on capabilities the field has begun but not finished building. We close by naming the open problems whose resolution would most advance embodied safety.

Verified safety envelopes for high-dimensional systems

Provable safety guarantees exist for simple, low-dimensional systems, but scaling them to the high-dimensional, contact-rich, perception-dependent settings of real robots and vehicles remains open. The prize is a supervisory layer that is both provably safe and not so conservative that it forecloses the capability the learned system was meant to provide. Bridging formal control-theoretic guarantees with learned perception is, in our view, the central technical problem of the field.

Reliable uncertainty and out-of-distribution detection

Principle three depends on a system's ability to know what it does not know. Current learned systems are notoriously poor at this — confidently wrong on inputs unlike their training data. Uncertainty quantification and out-of-distribution detection that are reliable enough to trigger safety responses, fast enough to fit real-time budgets, and calibrated enough to avoid crying wolf, remain unsolved and essential.

Closing and quantifying the sim-to-real gap

Simulation will remain the backbone of embodied testing, which makes the disciplined measurement of the sim-to-real gap a standing research need. Methods that quantify how far a simulator can be trusted for a given claim — and that transfer learned behavior to reality with bounded, characterized loss — would convert simulation from a source of possibly-false confidence into a rigorous validation instrument.

Human-factors science for shared autonomy

The handoff problem, automation complacency, and trust calibration are human-factors questions as much as engineering ones, and the science is far from settled for the specific case of learned,

embodied systems that are competent most of the time. Empirical understanding of how people actually behave in prolonged shared control — and design patterns that account for it — is a research frontier with direct safety payoff.

Governance instruments for learned physical systems

Finally, the governance apparatus itself is a research object. Certification methods appropriate to systems that cannot be exhaustively specified, liability doctrines fit for systems that learn and update, and standards that reach the perceptual and social domains as competently as they reach the mechanical ones — all remain works in progress, and all require collaboration between technologists, safety engineers, human-factors researchers, and policymakers that the field has only begun to organize.

CLOSING NOTE

Embodied AI is not a distant prospect; it is arriving now, across every domain this report surveys. The window in which its safety practices are established — before deployment scales past the point where they can be retrofitted — is open, and it is not indefinite. The disciplines that must contribute already exist: control theory knows how to bound dynamical systems, safety engineering knows how to govern physical risk, human-factors research knows how people and machines fail together, and machine learning supplies the capability. The task is to join them into a coherent practice before the systems they must govern become the infrastructure of daily life. That is the work this report is meant to help begin.

APPENDIX

Notes & Further Reading

This report synthesizes established work in robotics, control and safety engineering, human-robot interaction, and AI governance. The sources below are entry points into the literature that informs it; they are indicative rather than exhaustive.

1. Leveson, N. *Engineering a Safer World: Systems Thinking Applied to Safety.* On treating safety as a control problem across a sociotechnical system rather than a component property — foundational for reasoning about embodied loops.
2. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. *Concrete Problems in AI Safety.* On robustness, distributional shift, safe exploration, and avoiding negative side effects as engineering problems.
3. Ross, S., Gordon, G., & Bagnell, D. *A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning (DAgger).* On the compounding-error problem in sequential decision-making.
4. International Organization for Standardization. *ISO 10218 & ISO/TS 15066 — Robots and robotic devices: safety requirements and collaborative operation.* On power-and-force limiting and speed-and-separation monitoring for collaborative robots.
5. International Organization for Standardization. *ISO 26262 — Road vehicles: functional safety.* On hazard analysis, risk classification, and integrity requirements for automotive systems.
6. International Organization for Standardization. *ISO 21448 — Safety of the Intended Functionality (SOTIF).* On hazards arising from performance limitations rather than component faults — directly relevant to learned perception.
7. Koopman, P., & Wagner, M. *Challenges in Autonomous Vehicle Testing and Validation.* On the long tail, the limits of testing, and the operational design domain.
8. Parasuraman, R., & Riley, V. *Humans and Automation: Use, Misuse, Disuse, Abuse.* On automation complacency, trust calibration, and meaningful human oversight.
9. Endsley, M. R. *Toward a Theory of Situation Awareness in Dynamic Systems.* On situational awareness and its loss under automation — central to the handoff problem.
10. Dragan, A., Lee, K., & Srinivasa, S. *Legibility and Predictability of Robot Motion.* On designing motion that communicates intent to nearby humans.
11. Ames, A., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. *Control Barrier Functions: Theory and Applications.* On enforcing safety constraints as a supervisory envelope around a controller.
12. National Institute of Standards and Technology. *AI Risk Management Framework (AI RMF 1.0).* On organizing AI risk into govern, map, measure, and manage functions across the lifecycle.