Robots Need More Than VLAs & World Models

Elis Karcini Motoniq.ai
Faisal Mehrban Motoniq.ai
Nguyen Pham Motoniq.ai
Mac Schwager Stanford University, Motoniq.ai
Arash Ajoundani Istituto Italiano di Tecnologia
César Cadena ETH Zurich
Jan Peters Technical University of Darmstadt
Marco Hutter ETH Zurich
Haitham Bou-Ammar UCL Centre for AI

Abstract

Generalist robot intelligence is often framed as a policy-scaling problem: collect more robot demonstrations, train larger Vision-Language-Action (VLA) models, and expect broader generalisation. In this position paper, we argue that this framing is incomplete. The central bottleneck is not only policy learning, but the absence of mechanisms that convert the world's abundant unstructured behavioural data into grounded robot supervision. Human motion, internet video, simulation rollouts, and interactive demonstrations contain rich information about tasks, goals, contacts, failures, and physical constraints, yet most of this information is not directly usable by robot policies because it lacks embodiment-specific action labels, task semantics, and reward structure. We identify four missing components for the next generation of robotics: data interfaces for autolabelling unstructured behaviour, embodiment interfaces for retargeting human motion to robot actions, world-model interfaces for physics-grounded 3D reasoning, and reward interfaces for inferring task progress and success from video and language. We survey recent progress in robot foundation models, cross-embodiment datasets, learning from video, world models, and reward modelling, and propose a research agenda for building robotics systems that can learn not only from robot demonstrations, but from the broader physical world.

1 Introduction

Robotics is entering its foundation-model moment, but it does not yet have its internet. Large-scale vision-language-action models, cross-embodiment datasets, learned simulators, and interactive world models have made it increasingly plausible that robots will eventually acquire broad, reusable physical skills rather than being programmed task by task. Yet the path to generalist robotics remains far less clear than the path that enabled progress in language and vision. Text and images are abundant, naturally digitised, and densely associated with human-generated supervision. Physical interaction is different: the world contains vast amounts of behavioural data, but most of it is not directly usable by robots. Human demonstrations, internet videos, factory workflows, household activities, and simulation rollouts often reveal what task is being attempted, how objects move, when contact occurs, and whether an outcome succeeds or fails. However, they rarely provide the embodiment-specific action labels, force signals, task semantics, or reward structure required to train robot policies. The central bottleneck for future robotics is therefore not only how to scale policies, but how to convert unstructured physical experience into grounded robot supervision.

Based on NeurIPS template. Position Paper.

Recent progress has begun to address this bottleneck by scaling robot datasets, pooling experience across embodiments, and training generalist vision-language-action policies. These efforts are important: they show that robot behaviour can improve when models are exposed to more tasks, more environments, and more bodies. However, they also reveal a deeper limitation. Most current pipelines still depend on explicitly collected robot demonstrations, manually specified tasks, curated datasets, or embodiment-specific action spaces. This makes progress expensive and difficult to scale. A robot dataset is not like a text corpus: every trajectory must be physically executable, every action is tied to a particular body, and every failure may damage hardware, objects, or the environment. As a result, the amount of usable robot supervision remains tiny compared with the amount of physical behaviour already present in the world. The key question is therefore not only how to collect more robot data, but how to make broader sources of physical experience usable for robot learning.

In this position paper, we survey the rapidly growing body of work that attempts to widen the sources of supervision for robot learning, including vision-language-action models trained on heterogeneous robot trajectories, methods that extract behavioural priors from human and internet-scale video, action-conditioned world models for imagined robot experience, and simulation-based pipelines for generating counterfactual interaction data. Rather than organising this progress only by data source or algorithmic family, we argue that a more useful organising principle is the robotics learning pipeline itself. Today's pipeline is largely robot-data-centric: collect robot demonstrations, attach task or language labels, train a policy, evaluate on hardware, and repeat. We argue that the future pipeline must instead be grounding-centric: start from broad physical experience, e.g., human motion, internet video, robot interaction, simulation, tactile sensing, and language and pass it through grounding mechanisms that produce robot-usable actions, contacts, object states, task phases, goals, and rewards. The central question for the field is therefore not simply which model architecture to train, but which mechanisms are needed to make the world's physical experience learnable by robots. Importantly, our argument is not that VLA models are unimportant. Rather, VLAs should be understood as one layer in a larger physical-intelligence stack: a policy interface that depends on upstream grounding of data, embodiment, dynamics, rewards, and deployment feedback.

We therefore organise the survey not merely by model family, dataset, or algorithmic trend, but by the supervision bottleneck each line of work exposes. Robot-native datasets show how far policy learning can scale when actions and task labels are already available. Video-based methods show that the world contains abundant behavioural evidence, but weak grounding. Simulation and world models show how experience can be generated, but only when physical consequences are preserved. This organisation leads to our central claim: the missing layer in robotics is not another policy architecture alone, but a set of components that transform physical experience into robot-usable supervision.

We argue that the central question for the field is not simply which model architecture to train, but which mechanisms are needed to make the world's physical experience learnable by robots. We argue that this requires four missing pillars: a physical data engine for embodied autolabelling, task-preserving retargeting across embodiments, physics-grounded world models for consequence prediction, and task-conditioned reward grounding through deployment loops. These pillars provide the organising principle for the remainder of the paper: we first survey how current robot-learning systems expose the grounding bottleneck, and then outline the components needed to move from physical experience to physical intelligence.

2 Robot-Native Supervision: Progress and Scaling Limits

Much of contemporary robot learning remains organised around robot-native supervision. A robot is placed in an environment, demonstrations or interaction trajectories are collected, and the resulting observations are paired with embodiment-specific actions, task labels, language instructions, or success signals. This paradigm has enabled substantial progress in imitation learning, reinforcement learning, and vision-language-action modelling, especially as datasets have expanded across tasks, environments, and robot embodiments. At the same time, the field is already searching for ways to move beyond this regime: simulation is used to generate scalable experience, real-to-sim-to-real methods aim to amplify scarce real data, human and internet videos are used to learn behavioural priors, and world models are trained to support prediction, planning, and counterfactual reasoning. Our point is therefore not that robotics is limited to one pipeline, but that much of today's usable supervision still becomes useful only after it has been grounded into robot-native quantities. The

central scaling limit is this grounding step: how do we turn broader physical experience into actions, contacts, object states, task phases, goals, and rewards that a robot can learn from?

2.1 The Robot-Native Regime

By robot-native supervision, we mean physical experience that is already represented in the coordinate system of a robot-learning problem. In the standard case, this consists of trajectories collected from a particular embodiment, where robot observations are paired with robot actions and, in some cases, language instructions, task labels, rewards, or success indicators. The observations may include camera images, proprioceptive states, end-effector poses, tactile readings, force-torque measurements, or other sensor streams; the actions may correspond to joint commands, end-effector displacements, gripper states, velocity commands, or higher-level skill primitives. This form of data is powerful because it directly matches the supervised or reinforcement-learning objective: a model can imitate the demonstrated action, optimise the provided reward, or condition behaviour on the associated task description. In other words, the data is useful because it has already been grounded into a particular robot body, action space, sensor suite, and task definition.

This regime has enabled a series of increasingly ambitious efforts to scale robot learning through larger and more diverse robot datasets. Early multi-robot datasets such as RoboNet demonstrated the value of sharing robotic experience across platforms, containing 15 million video frames from seven robot platforms and supporting both video-prediction and inverse-model learning [Dasari et al., 2019]. More recent datasets have expanded the scale and diversity of robot-native supervision: BridgeData V2 [Walke et al., 2023] provides roughly sixty thousand manipulation trajectories across twenty-four environments on a low-cost robot platform, while DROID contains approximately seventy-six thousand demonstration trajectories, or 350 hours of interaction data, collected across hundreds of scenes and dozens of tasks by geographically distributed data collectors [Khazatsky et al., 2024]. RH20T further broadens this trend by collecting over 110,000 contact-rich manipulation sequences with visual, force, audio, and action information, together with corresponding human demonstration videos, making it especially relevant for studying multimodal grounding in manipulation [Fang et al., 2023]. These datasets make clear that robot-learning performance improves when models are exposed to more tasks, objects, environments, and embodiments. They also show that data diversity is not a detail but a central requirement for generalisation beyond a single laboratory setup.

In parallel, robot-native trajectories have become the substrate for increasingly generalist robot policies. BC-Z studied how scaling real-robot imitation data across more than one hundred tasks can enable zero-shot generalisation to unseen manipulation tasks, including conditioning policies on language or videos of humans performing the task [Jang et al., 2022]. RT-1 showed that a transformer policy trained on approximately 130,000 real-robot episodes collected across thirteen robots and more than 700 tasks could produce broad language-conditioned manipulation behaviour [Brohan et al., 2022]. RT-2 extended this direction by co-training vision-language models on web-scale vision-language data and robot trajectories, representing robot actions as tokens so that semantic knowledge from internet-scale pretraining could be transferred into robotic control [Zitkovich et al., 2023]. SayCan [Ahn et al., 2022] and PaLM-E [Driess et al., 2023] similarly illustrate how large language or multimodal models can contribute to semantic reasoning and planning, but still require grounding through robot affordances, embodied observations, or learned low-level skills.

The same trend appears in cross-embodiment and open-source generalist policy efforts. Open X-Embodiment and RT-X pooled more than one million real robot trajectories from 22 robot embodiments by aggregating datasets from many research laboratories into a common format, making cross-embodiment training a practical research direction [O'Neill et al., 2024]. Octo then showed that an open-source generalist policy can be pretrained on 800,000 trajectories from Open X-Embodiment and adapted to new observation and action spaces [Team et al., 2024], while RoboCat [Bousmalis et al., 2023] explored a self-improving generalist manipulation agent trained on action-labelled experience across multiple robots and tasks. Systems such as Dobb-E further push robot learning into less curated settings by collecting household demonstrations and adapting policies for new home tasks, highlighting both the promise and the messiness of real-world deployment [Shafiullah et al., 2023]. Taken together, these works mark a major shift from single-task, single-robot learning toward broader robot foundation models. Yet they also reinforce the central premise of this section: the most effective supervision still largely arrives as robot-grounded trajectories, where actions, observations, task descriptions, and success signals have already been made legible to a robot-learning algorithm.

A complementary line of work has improved the policy-learning machinery applied to robot-native trajectories. Diffusion Policy, for example, formulates visuomotor control as conditional denoising over action sequences, showing that diffusion models can represent multimodal action distributions and produce strong manipulation policies from demonstration data [Chi et al., 2025]. ALOHA [Fu et al., 2024, Zhao et al., 2024] and Action Chunking with Transformers [George and Farimani, 2023, Bharadhwaj et al., 2024, Zhang, 2025] similarly show that carefully designed low-cost teleoperation systems, combined with sequence-level imitation learning, can acquire fine-grained bimanual manipulation skills from real-world demonstrations

Those developments have contributed to a rapid expansion of vision-language-action models, i.e., models that map observations and textual descriptions to robotic actions. Early systems such as Gato [Reed et al., 2022] helped establish the idea that a single transformer-style agent could operate across modalities and domains, including real robot control. At the same time, SayCan [Ahn et al., 2022] and PaLM-E [Driess et al., 2023] showed how large language or multimodal models could support embodied reasoning, planning, and affordance-aware skill selection, provided they were grounded through robot skills, observations, or low-level policies. RT-2 then made the VLA framing more explicit by co-training web-scale vision-language models with robot trajectories and representing robot actions as tokens, allowing semantic knowledge from internet-scale pretraining to be transferred into robotic control [Zitkovich et al., 2023]. Related systems explored different forms of multimodal task specification and VLM adaptation: VIMA formulates robot manipulation as multimodal prompting over interleaved text and visual tokens, while RoboFlamingo adapts open vision-language foundation models for language-conditioned robotic manipulation [Jiang et al., 2023]. Moreover, OpenVLA is a representative open-source example: it trains a 7B-parameter VLA on approximately 970,000 real-world robot demonstrations from Open X-Embodiment, taking images and language instructions as input and producing robot actions as output [Kim et al., 2024]. Physical Intelligence's π_0 similarly frames generalist robot control as a vision-language-action problem, using a flow-matching architecture built on top of a pretrained vision-language model to inherit internet-scale semantic knowledge while producing continuous robot actions [Black et al., 2024]. A related line of work focuses on the action-generation mechanism itself: CogACT separates the vision-language reasoning component from a specialised action module and studies diffusion action transformers for action-sequence modelling, while RoboMamba explores state-space-model architectures for efficient vision-language-action reasoning and manipulation [Li et al., 2024, Liu et al., 2024a]. Furthermore, FAST and related action-tokenisation methods address a complementary bottleneck: how high-frequency continuous robot actions should be compressed or tokenised so that they can be modelled efficiently by VLA architectures [Moodley et al., 2024, Zhong et al., 2025, Pertsch et al., 2025, Liu et al., 2025a, Dong et al., 2026].

Other recent models have focused on more specialised ingredients needed for scalable VLA control: SpatialVLA incorporates explicit spatial representations for robot manipulation and is trained on approximately 1.1 million real robot episodes, while RDT-1B [Liu et al., 2024b] uses a diffusion-transformer architecture for bimanual manipulation and is pretrained on more than one million multi-robot episodes [Zhang et al., 2024, Qu et al., 2025b, Patratskiy et al., 2025, Feng et al., 2025]. This spatial focus is also reflected in 3D-VLA, which argues that robot foundation models should move beyond 2D visual inputs by linking 3D perception, embodied reasoning, action prediction, and generative world modelling [Qu et al., 2025a]. Related spatially grounded VLA efforts include 3DS-VLA, which uses 3D spatial constraints and point-cloud information for manipulation [Li et al., 2025b]; GeoVLA, which integrates depth-derived point clouds through a point encoder and a 3D-enhanced action expert [Sun et al., 2025]; GraphCoT-VLA, which introduces a 3D pose-object graph and structured chain-of-thought reasoning for ambiguous manipulation instructions [Huang et al., 2026a]; and Avi, which reframes robotic action generation as a problem of 3D perception and language-grounded spatial reasoning rather than only low-level policy learning [Song and Le, 2025]. Another emerging concept for endowing robot policies with strong spatial-physical priors is to build policies on top of video generation backbones. One such paradigm is the World -Action Model (WAM), which predicts video frames as well as action chunks. DreamZero Ye et al. [2026] and Unified Video Action (UVA) Model Li et al. [2026] are both prime examples.

More recent humanoid-focused systems extend the VLA paradigm from tabletop manipulation toward whole-body and dexterous control. NVIDIA Isaac GR00T N1 explicitly targets generalist humanoid robots and is described as a dual-system VLA: a vision-language module interprets the scene and instruction, while a diffusion-transformer action module generates real-time motor

commands for humanoid control [Bjorck et al., 2025]. Importantly, GR00T N1 is trained on a mixture of egocentric human videos, real and simulated robot trajectories, and synthetic data, making it a useful example of the broader data mixture now being explored for humanoid robotics. Gemini Robotics follows a similar foundation-model direction, presenting a generalist VLA built on Gemini that can directly control robots from visual observations and language instructions, with later on-device variants reported to support deployment without constant internet connectivity and adaptation to other platforms such as Apptronik's Apollo and Franka FR3 [Team et al., 2025]. Figure's Helix is another humanoid-focused VLA system, presented as a model that unifies perception, language understanding, and learned control for full upper-body humanoid manipulation, including wrists, torso, head, and fingers [Figure AI, 2025]. Beyond these frontier model releases, recent research has begun to study humanoid-specific VLA structure more directly. LeVERB proposes a hierarchical VLA framework for humanoid whole-body control, learning a latent action vocabulary from synthetically rendered kinematic demonstrations and using a reinforcement-learned whole-body controller to execute dynamics-level commands [Xue et al., 2025]. WholeBodyVLA similarly targets closed-loop humanoid loco-manipulation, learning unified latent actions from action-free egocentric videos and combining them with a locomotion-oriented control policy [Jiang et al., 2025]. HuMI addresses the data-collection bottleneck from a different angle, using portable robot-free demonstrations to learn whole-body humanoid manipulation skills and reporting improved data-collection efficiency compared with teleoperation [Nai et al., 2026]. HEX focuses on cross-embodiment whole-body manipulation, introducing humanoid-aligned state representations and expert modules to improve coordinated control on full-sized bipedal robots [Bai et al., 2026]. Industry systems such as Skild Brain also frame humanoid and general-purpose robot control as an omni-bodied foundation-model problem trained from simulation, human action videos, and real-world robot feedback [Skild AI, 2025].

The breadth of this literature shows how far robot learning has moved beyond single-task, single-platform imitation. Yet the underlying supervision regime remains largely robot-native. The strongest results still depend on experience that has already been expressed as observations paired with actions, tasks, rewards, demonstrations, or success labels. This is precisely why these systems work, but it also identifies the scaling bottleneck: most of the physical behaviour available in the world does not arrive with explicit robot actions.

Takeaway. Robot-native supervision has enabled the most impressive progress in generalist robot policies. However, its strength is also its limitation: the data has already been expressed in the coordinate system of robot learning. Actions, task labels, embodiment constraints, and success signals are either collected directly or curated afterwards. This makes VLA scaling powerful, but still dependent on supervision that has already been grounded.

2.2 Learning from Weakly Grounded Physical Observations

To enable *true scalability*, a growing body of work asks whether robot learning can benefit from physical observation that is not natively action-labelled. This direction is motivated by a simple asymmetry: robot action-labelled trajectories are expensive to collect, but videos of humans and physical interactions are abundant. Human videos contain information about object affordances, task structure, contact events, temporal progress, and failure recovery, even when they do not specify the motor commands that a robot should execute. The central question is whether such passive physical observations can be converted into useful learning signals for robot policies.

A useful way to formalise the problem is to distinguish between observed physical change and executable robot action. A video provides an observation sequence: $\mathbf{o}_{1:\mathsf{T}} = \langle o_1, \ldots, o_\mathsf{T} \rangle$, but not the robot action sequence: $\mathbf{a}_{1:\mathsf{T}} = \langle a_1, \ldots, a_\mathsf{T} \rangle$. Robot-native imitation learning assumes access to pairs (o_t, a_t), while learning from human or internet videos usually provides only $\mathbf{o}_{1:\mathsf{T}}$, sometimes with language $\mathrm{L}_{1:\mathsf{T}}$, captions, or weak task metadata. The latent variables are therefore action-like representations $\mathbf{z}_{1:\mathsf{T}}$ that explain transitions from o_t to o_{t+1} such that: $z_t \sim q(\cdot|o_t, o_{t+1}, \mathrm{L}_t, \mathrm{L}_{t+1})$. Generally, those latent variables are not yet tied to any particular robot embodiment. The hope is that such latent actions capture task-relevant changes in the world, e.g., moving, grasping, opening, placing, inserting, aligning, and can later be mapped to embodiment-specific robot actions. This makes latent-action learning a natural bridge between passive video and robot-native control.

Representation Learners. Several representation-learning works can be viewed as indirect steps in this direction. R3M pretrains visual representations on the Ego4D human video dataset using time-contrastive learning, video-language alignment, and sparsity regularisation, then uses the frozen representation for downstream robot manipulation policies [Nair et al., 2022]. VIP learns visual representations from human videos by using temporal distance as a proxy for task progress, producing features that can support robotic reinforcement learning and imitation [Ma et al., 2022]. MVP studies masked visual pretraining for robot manipulation, while VC-1 evaluates a range of pretrained visual representations across robotic tasks and argues for large-scale visual pretraining as a reusable perceptual substrate for embodied control [Radosavovic et al., 2023, Majumdar et al., 2023]. These methods do not directly recover robot actions from video, but they show that passive human or internet-scale visual experience can improve the perceptual and task-relevant features used by robot policies. R3M, in particular, explicitly studies how human video pretraining can enable data-efficient manipulation learning. However, these representation-learning methods mostly do not fully resolve the correspondence problem: how an observed physical change in human or internet video becomes a reward, subgoal, latent action, or executable control signal for a particular robot.

Earlier work on imitation from observation and cross-embodiment learning exposed the same weak-grounding problem in a more explicit form. Time-Contrastive Networks learned viewpoint-invariant representations from unlabeled multi-view videos and used the resulting embeddings for robotic imitation and reward learning, including imitation from human demonstrations without explicit action correspondence [Sermanet et al., 2018]. AVID addressed the human-to-robot appearance gap by translating human demonstration videos into robot-domain visual instructions, which were then used as rewards for model-based reinforcement learning [Smith et al., 2020]. XIRL formulated the problem as cross-embodiment inverse reinforcement learning, learning vision-based rewards from videos of agents with different bodies, actions, and end-effector dynamics [Zakka et al., 2021]. Similarly, DVD learned generalisable reward functions from a mixture of in-the-wild human videos and a small amount of robot data, using functional similarity between human and robot behaviour as the supervision signal [Chen et al., 2021]. Together, these works show that the core obstacle is not merely the absence of robot actions, but the broader correspondence problem: how to align observed physical progress with what a different robot body can perceive, value, and execute.

Latent-Action Approaches. More recent latent-action approaches attack the missing-action-label problem more directly. Latent Action Pretraining (LAPA) proposes an unsupervised method for pretraining VLA models without ground-truth robot action labels [Ye et al., 2024]. It first learns a discrete latent action space from video transitions using a VQ-VAE-style objective, then trains a latent VLA model to predict these latent actions from observations and task descriptions, before fine-tuning on smaller robot datasets to map latent actions to executable robot actions. This is particularly relevant to our thesis because LAPA treats internet-scale video not merely as semantic or perceptual pretraining data, but as a source of action-like structure. In other words, it takes the first steps to ask whether the physical changes visible in the video can be compressed into a reusable action vocabulary before being grounded into a particular robot body. A related line of work learns action-like abstractions from heterogeneous embodiments and viewpoints. UniVLA, for example, derives task-centric latent actions in an unsupervised manner, allowing the model to leverage data from arbitrary embodiments and camera perspectives without requiring action labels [Bu et al., 2025]. Strictly speaking, however, such latent variables are better understood as transition codes or physical-change descriptors until they are grounded in a specific robot embodiment. They become robot actions only when an embodiment-conditioned decoder can map them to commands that, when executed, reproduce the intended physical change.

Task-Progress Signals. Another branch uses video-language models to infer task progress, rewards, and success from passive observation. Rather than asking videos to provide actions, these methods ask videos to provide supervision over *what matters*. PROGRESSOR learns a task-agnostic reward function from unlabelled videos and uses self-supervised refinement to provide dense rewards for goal-conditioned policy learning [Ayalew et al., 2025]. Adapt2Reward transfers video-language models into language-conditioned reward functions using limited robot video data [Yang et al., 2024b]. ReWiND trains a reward model to predict video progress rewards from image or rollout embeddings and language instructions, using video rewinding and misaligned video-language pairs as negative supervision [Zhang et al., 2025a]. TimeRewarder derives progress signals from passive videos, including both robot demonstrations and human videos, by modelling temporal distances

between frame pairs [Liu et al., 2025b]. Stage-Aware Reward Model (SARM) uses dense subtask labels to help supervised a reward model to determine fine-grained task progress Chen et al. [2026]. These works are crucial because they suggest that passive video may be useful not only for perceptual pretraining or latent actions, but also for grounding task progress and reward.

This literature reveals a useful taxonomy of weak physical supervision. Passive videos can provide at least four kinds of signal. First, they can provide visual representations, as in R3M, VIP, MVP, and VC-1. Second, they can provide latent action codes, as in LAPA and UniVLA. Third, they can provide task-progress and reward signals, as in PROGRESSOR, Adapt2Reward, ReWiND, TimeRewarder, and SARM. Fourth, they can provide behavioural priors about object use, affordances, contact, and temporal task structure. Recent surveys on learning from human or internet video emphasise this same opportunity while highlighting the unresolved challenges: videos lack robot actions, viewpoints and embodiments differ, physical contact and forces are often unobserved, and human strategies may not be directly executable by a robot [Eze and Crick, 2025, Feng et al., 2026].

However, weak physical supervision does not remove the need for grounding; it relocates it. A latent action learned from videos is not yet a robot command. A progress signal inferred from temporal order is not necessarily a reward for a new embodiment. A visual representation trained on human videos may encode objects and affordances, but not the contact dynamics or force constraints needed for manipulation. Thus, the central challenge is not simply to pretrain on more video. It is to determine which variables should be extracted from video, how those variables should be grounded into robot morphology and control, and how errors in this grounding affect downstream policy learning. In this sense, learning from action-free video is one of the clearest examples of the broader thesis of this paper: the world contains abundant physical experience, but robotics still lacks reliable mechanisms for transforming that experience into robot-usable supervision.

Takeaway. Passive video can provide representations, progress signals, latent actions, and behavioural priors. But these signals are not yet robot supervision. A latent action is not a command; a progress signal is not necessarily a reward; and a human strategy may not be executable by a robot. Video expands the source of physical experience, but it also makes the grounding problem unavoidable.

2.3 Generating Physical Experience

A second response to the cost of robot-native supervision is not to infer labels from existing observations, but to generate additional physical experience. If robot trajectories are expensive because they must be enacted on hardware, then simulation, synthetic demonstration generation, and learned world models offer a complementary route: they can expose policies to more tasks, objects, initial conditions, failures, and counterfactual outcomes than would be practical to collect directly in the real world. This shifts the data-scaling question from "How many robot trajectories can we collect?" to "How faithfully can we generate experience that preserves the physical structure needed for control?" In this view, simulation and world models are not merely data factories; they are mechanisms for expanding the reachable distribution of physical interactions. Their value depends on whether the generated experience captures the quantities that matter for downstream learning: geometry, object state, contact, dynamics, embodiment constraints, task progress, and success or failure.

The Simulation Route. A first route to generating physical experience is simulation. Simulation environments and benchmarks provide controllable worlds in which tasks, objects, layouts, initial states, and demonstrations can be generated at a scale that would be difficult to reproduce on hardware. RLBench is an early and influential example, providing 100 hand-designed manipulation tasks, multi-modal observations, language descriptions, and an effectively unlimited supply of demonstrations generated through motion planning [James et al., 2019]. Meta-World [Yu et al., 2020] similarly helped standardise multi-task and meta-reinforcement learning for robotic manipulation, while ManiSkill focuses on generalisable manipulation from 3D visual inputs in a full-physics simulator with diverse object geometries [Mu et al., 2021]. CALVIN extends this direction toward long-horizon language-conditioned manipulation, asking agents to compose many behaviours from language instructions in simulated environments [Mees et al., 2022]. LIBERO further studies lifelong robot learning, knowledge transfer, and task-ordering effects across 130 language-conditioned manipulation tasks [Liu et al., 2023]. Together, these environments have been crucial because they make robot learning reproducible, scalable, and comparable across methods. Yet they also illustrate a recurring assump-

tion: the simulator designer has already specified the relevant state, action space, task definitions, object assets, and success conditions.

The Data Generation Route. A second route is to use simulation not only as an evaluation environment, but as a data-generation engine. MimicGen is a particularly important example: it automatically synthesises large-scale demonstration datasets in a simulator from a small number of human demonstrations by adapting demonstration segments to new object poses and contexts, generating more than 50,000 demonstrations from fewer than 200 seed demonstrations across 18 tasks [Mandlekar et al., 2023]. RoboCasa scales this idea toward everyday household manipulation by providing a large-scale simulated kitchen environment for training generalist robots, and its authors report clear scaling trends from using synthetically generated robot data for imitation learning [Nasiriany et al., 2024]. RoboCasa365 pushes this further with 365 everyday household tasks, 2,500 kitchen scenes, and over 2,000 hours of robot interaction data, including both human demonstrations and synthetic demonstrations generated with MimicGen [Nasiriany et al., 2026]. RoboGen takes a more automated route, using foundation and generative models to construct tasks, scenes, and training data in simulation so that robots can acquire diverse skills at scale [Wang et al., 2024]. These systems are important because they shift robot data collection from "manually collect every trajectory" to "collect a small amount of seed experience, then generate variations." The open question is whether those variations preserve the physical details that matter for real control, especially contacts, object stability, friction, deformation, and failure modes.

The Real-to-Sim-to-Real Route. Instead of relying on a hand-built simulator, these methods attempt to reconstruct or approximate a real environment, use simulation to expand or robustify learning, and then return the resulting policy to the physical world. RialTo is a representative example: it constructs digital-twin simulation environments from small amounts of real-world data and uses reinforcement learning in simulation to robustify imitation policies before real-world deployment [Torne et al., 2024]. More recent work has begun to use 3D Gaussian Splatting and related reconstruction methods to make digital twins more visually faithful and easier to build from real observations. RL-GSBridge introduces a 3D-Gaussian-Splatting-based real-to-sim-to-real reinforcement-learning framework for robotic manipulation, using reconstructed scenes to support zero-shot sim-to-real transfer for vision-based control [Wu et al., 2025]. Real-is-Sim similarly uses a dynamic digital twin based on Embodied Gaussians throughout data collection, training, policy evaluation, and deployment, aiming to reduce the gap between offline policy evaluation and real-world success [Abou-Chakra et al., 2025]. Related real-to-sim evaluation work constructs soft-body digital twins from real-world videos and renders robots, objects, and environments with photorealistic fidelity using 3D Gaussian Splatting, showing that simulated rollouts can correlate with real policy performance on deformable manipulation tasks such as plush packing and rope routing [Zhang et al., 2025b]. RoboGSim also follows this direction, presenting an interactive real2sim2real Gaussian-splatting platform for demonstration synthesis, novel-scene and novel-object data scaling, and closed-loop policy evaluation [Li et al., 2025c]. For robot navigation tasks in particular, policies trained in 3D Gaussian Splatting simulators have been shown to transfer zero-shot to real world deployment. SOUS VIDE and SINGER are examples of such approaches using an imitation learning paradigm Low et al. [2025], Adang et al. [2026], while GRaD-Nav and GRaD-Nav++ train with RL and exploit end-to-end differentiability of 3DGS rendering Chen et al. [2025b,a].

This direction builds on a broader sim-to-real literature that has long studied how generated experience can transfer to physical robots despite imperfect simulators. Domain randomisation is one of the dominant strategies: rather than fitting a single simulator, it trains policies over a distribution of simulated dynamics or visual parameters, so that the real world appears as a single variation within the training distribution [Muratore et al., 2022]. Those ideas have also been used in legged robotics: learning agile and dynamic motor skills for ANYmal showed that neural policies trained in simulation can transfer to a real quadruped, while later massively parallel reinforcement-learning work showed that locomotion policies can be trained in minutes in simulation and transferred to real robots [Hwangbo et al., 2019, Rudin et al., 2022]. Domain randomisation in simulation can also be combined with online latent parameter adaptation to better facilitate the sim-to-real transfer, for example, using the Rapid Motor Adaptation (RMA) approach for legged locomotion Kumar et al. [2021], and further adapted to manipulation tasks in Wang et al. [2026a]. These works are not always "real-to-sim-to-real" in this sense. Still, they are essential context because they show why simulation became attractive in the first place: it generates experience cheaply, safely, and at scale, but only transfers when the simulator captures or randomises the physical factors that matter.

The World-Modelling Route. A fourth route is to replace or augment explicit simulation with learned world models. The idea has a long history: rather than learning only a reactive policy, an agent can learn an internal predictive model of how the environment changes under its actions and then use that model for planning, imagination, or policy improvement. Schmidhuber's early work on "making the world differentiable" already proposed reinforcement learning and planning with recurrent neural networks, including a controller and a learned world model that predicts environmental dynamics [1]. This view was later popularised in deep learning by Ha and Schmidhuber's World Models, which trained a generative recurrent model of the environment and then learned compact policies from the model's latent representations, including policies trained inside the model's own "dreamed" rollouts before being transferred back to the real environment [Ha and Schmidhuber, 2018a,b]. In modern reinforcement learning, PlaNet and Dreamer made this idea practical at scale by learning compact latent dynamics from pixels and improving policies through imagined futures; DreamerV3 later showed that a single world-model algorithm can solve a wide range of control tasks with fixed hyperparameters, while DayDreamer demonstrated that Dreamer-style world models can be learned directly on physical robots for locomotion, manipulation, and navigation without relying on a hand-built simulation [Hafner et al., 2019b,a, 2020, Wu et al., 2022, Hafner et al., 2025].

In robotics, this tradition is being extended from latent dynamics for control toward models that can generate or predict physically meaningful experience. RoboDreamer learns compositional video world models for robot imagination by factorising video generation according to language-derived primitives, allowing it to synthesise plans for unseen combinations of objects and actions [Zhou et al., 2024]. UniSim asks whether a universal interactive simulator can be learned from diverse datasets, combining information from images, robotics data, and navigation data to simulate the visual outcomes of high-level instructions and low-level controls; it also shows that policies trained in the learned simulator can transfer to real-world deployment in some settings [Yang et al., 2024a]. DeepMind's Genie extends this line toward generative interactive environments trained from unlabelled internet videos, introducing a spatiotemporal tokeniser, an autoregressive dynamics model, and a learned latent action model that enables frame-by-frame control without ground-truth action labels [Bruce et al., 2024]. These systems are exciting because they blur the boundary between video generation, simulation, and robot learning: generated experience is no longer just passive video, but potentially an action-conditioned environment in which agents can plan or train.

However, robotics places stricter demands on world models than visual plausibility alone. A useful robot world model must preserve the variables that matter for action: 3D geometry, object permanence, contact, material properties, constraints, forces, and the consequences of robot motion. This has motivated a growing line of *3D, object-centric, and physics-grounded world models*. Object-centric world models such as FOCUS argue that manipulation requires representing objects and their interactions explicitly, enabling more efficient exploration of robot-object dynamics [Ferraro et al., 2023]. Related language-guided object-centric world models predict future states in compact object-centric representation spaces rather than only generating pixels, improving efficiency for visuo-linguo-motor control [Jeong et al., 2025]. PointWorld scales this idea to 3D by unifying state and action in a shared spatial domain and predicting full-scene 3D point flow from RGB-D observations and robot actions [Huang et al., 2026b]. ParticleFormer similarly learns a transformer-based 3D point-cloud world model for multi-object, multi-material manipulation, predicting dynamics directly from real-world robot perception data and supporting downstream manipulation through model-predictive control [Huang et al., 2025]. These methods are especially relevant because they move world modelling away from generic image prediction and toward spatially grounded, action-conditioned dynamics.

When developing world models for robot imagination, planning, data generation, or evaluation, it is critical to assess the confidence of a world model prediction. Learned world models, as all learned models, are subject to errors and hallucinations when queried outside the distribution of their training data. For a robot using a world model for planning, this can lead to a viscous cycle, where a hallucination leads to a poor control choice, leading the world model further away from its training distribution invoking further hallucinations, which further degenerate control effectiveness. Therefore, for robotics in particular, world models need to have a calibrated estimate of the certainty of their own predictions. Early work on this topic includes Mei et al. [2025], which learns a latent uncertainty quantification with a VAE approach, statistically calibrated to yield interpretable uncertainties. Latent uncertainties can then be passed to the pixel level for visualisation of uncertain regions of the predicted

[1] `https://people.idsia.ch/~juergen/world-models-planning-curiosity-fki-1990.html`.

image. The authors of Li et al. [2025a] further show the importance of uncertainty quantification for a world model used as an environment for training reinforcement learning policies. In Ward et al. [2026] the authors show a world model with calibrated uncertainty in the latent space can be used to detect runtime errors in a VLA manipulation policy. Uncertainty quantification for world models is still a young area, but we anticipate work on this topic to grow as world models become more integrated into robot autonomy stacks.

Another way to prevent erroneous world model predictions is to structure a world model as a combination of a neural scene representation with physical simulation. This can be viewed as a modern, 3D neural-scene version of a much older model-based robot-learning idea: learn or construct a predictive model of the world, then use it for planning, policy optimisation, or data-efficient control. Classical robot-learning work already emphasised this model-based view [Kober et al., 2013] and discussed the importance of models, data efficiency, exploration, and safe real-world learning in robotics. PILCO [Deisenroth and Rasmussen, 2011] and its variants [Polymenakos et al., 2019, Cowen-Rivers et al., 2022] are another canonical example, using probabilistic Gaussian-process dynamics models to perform data-efficient policy search with uncertainty-aware long-term predictions.

A more structured version of this model-based view attempts to ground learned dynamics directly in physical laws or relational structure, rather than treating the transition model as an unconstrained black box. Deep Lagrangian Networks impose the structure of Lagrangian mechanics on neural models of robot dynamics, learning physically plausible inertia, Coriolis, gravitational, and control-dependent terms for model-based control [Lutter et al., 2019]. Hamiltonian Neural Networks learn a Hamiltonian function and use Hamilton's equations to produce energy-aware dynamics [Greydanus et al., 2019], while Lagrangian Neural Networks parameterise a Lagrangian and derive equations of motion through the Euler-Lagrange equations [Cranmer et al., 2020]. Related geometric dynamics models, such as Symplectic ODE-Net, enforce Hamiltonian or symplectic structure to improve long-horizon physical prediction [Zhong et al., 2019]. A complementary family represents the world as interacting objects, particles, or relations: Interaction Networks introduced graph-based reasoning over objects and relations for physical prediction [Battaglia et al., 2016], Neural Physics Engines model physical systems through learned object-centric interactions [Chang et al., 2017], and Graph Networks as Learnable Physics Engines showed that graph networks can simulate complex physical systems and generalise across different numbers and configurations of objects [Sanchez-Gonzalez et al., 2018]. Later graph-network simulators scaled this idea to particle-based fluids, rigid materials, and deformable systems [Sanchez-Gonzalez et al., 2020]. These works are important because they show that a robot world model can be grounded not only by more data, but also by inductive biases that encode energy, geometry, object relations, constraints, and interaction structure.

Of course, the contemporary versions of those ideas increasingly use learned visual and geometric representations as the substrate of the world model. LeCun's "A Path Towards Autonomous Machine Intelligence" argues that autonomous agents need predictive world models for planning under uncertainty, and the JEPA family operationalises this vision by predicting in abstract representation spaces rather than directly reconstructing pixels [LeCun et al., 2022]. I-JEPA introduced image-based joint-embedding predictive learning, while V-JEPA extended this to video by predicting masked parts of videos in representation space rather than pixel space [Assran et al., 2023]. V-JEPA 2 is especially relevant here because it combines internet-scale video with a small amount of robot interaction data and reports prediction, planning, and zero-shot robot control capabilities, making it one of the clearest recent links between JEPA-style world models and embodied control [Assran et al., 2025].

A further recent strand combines neural scene representations with physical simulation, aiming to build world models that are both visually grounded and physically actionable. Physically Embodied Gaussian Splatting proposes a dual Gaussian-particle representation that couples visual rendering with particle-based physical prediction and online correction based on observations, enabling the model to reason about present and future physical states while staying synchronised with the real world [Xie et al., 2024]. Gaussian World Models similarly use 3D Gaussian Splatting as a dynamic world representation for robotic manipulation, supporting action-conditioned 3D video prediction, imitation-learning representations, and model-based reinforcement learning [Lu et al., 2025]. ContactGaussian-WM pushes this direction toward contact-rich manipulation by learning a physics-grounded rigid-body world model from sparse contact-rich videos, combining a unified Gaussian representation for visual appearance and collision geometry with differentiable contact dynamics and closed-form physical reasoning; it reports applications to data synthesis and real-time model-predictive control [Wang et al.,

2026b]. Physics-informed world models for non-prehensile or deformable manipulation similarly attempt to inject physical structure into learned prediction rather than relying on unconstrained video generation, for example, by combining differentiable physics, visual observations, and physics-aware randomisation for robust sim-to-real manipulation [of PIN-WM, 2025]. These approaches make explicit what many video world models leave implicit: for robot planning, the generated future must obey enough geometry, contact, dynamics, and physical constraints to be useful for control.

Taken together, these strands suggest that world models for robotics should not be evaluated only by how realistic their generated observations appear. Their central purpose is to make physical experience counterfactual. A robot should be able to ask what would happen if it pushed at a different point, grasped with a different orientation, opened a drawer further, inserted an object at a slightly different angle, or stopped applying force. In a robot-native dataset, only the executed trajectory is observed; in a world model, the agent can imagine alternatives. But this advantage is meaningful only if the imagined futures preserve the variables that determine success and failure: object state, geometry, contact, force, stability, material response, embodiment constraints, and task progress. A visually plausible prediction that ignores contact, mass, friction, or physical feasibility may help representation learning, but it is not yet a reliable substrate for robot control. Thus, learned world models sharpen rather than solve the grounding problem: they promise scalable generated experience, but only if the generated experience is physically grounded and actionable.

Takeaway. Generated experience is useful only when it preserves the physical variables that determine control. A visually plausible rollout that ignores contact, force, friction, or stability is not yet reliable robot supervision. The value of simulation and world models is therefore not visual realism alone, but physically grounded counterfactual experience.

3 The Missing Components for Physical Intelligence

The survey above suggests that the next step in robotics is not simply to train larger policies, collect more demonstrations, or build more visually realistic simulators. These directions are necessary, but *incomplete.* What is missing is a set of components that transform broad physical experience into grounded, deployable robot intelligence. Today, robot learning often begins once the relevant variables have already been specified: observations, actions, task labels, rewards, success metrics, and embodiment constraints. Future physical-intelligence systems must instead recover these variables from weaker, messier, and more diverse sources of experience: human motion, internet video, wearable sensing, tactile streams, robot rollouts, simulation, language, and failure traces. In this sense, the central challenge is: how should a robot-learning system be organised so that the world itself becomes a source of supervision? We argue that such a system requires four missing components. First, it needs a physical data engine that can ingest heterogeneous experience and convert it into structured signals such as object states, contact events, task phases, latent actions, and success or failure labels. Second, it needs task-preserving retargeting, so that human behaviour and inferred skills can be translated into executable actions across different robot embodiments. Third, it needs physics-grounded world models that predict not only plausible future observations, but the physical consequences of action: geometry, contact, force, stability, constraints, and material response. Fourth, it needs a self-improving deployment loop, where each robot deployment produces new data, new failures, and new corrections that compound into broader competence rather than remaining isolated engineering effort. The proposed stack should therefore be understood as closed-loop rather than purely feed-forward. Rather than treating deployment only as evaluation, a physical-intelligence system should turn real rollouts, failures, and human corrections into structured supervision. Once grounded into contacts, object-state changes, reward errors, or failure modes, these traces can update the policy, reward model, world model, or retargeting model, allowing robots to improve on the actual tasks they face.

We believe that the next foundation model for robotics will not be only a VLA or a world-model. It will be a pipeline that grounds physical experience into actions, rewards, world models, and deployment feedback.

3.1 Physical Data Engines and Embodied Autolabelling

The first missing component is a physical data engine: a system that turns heterogeneous physical experience into structured learning signals for robots. Today, much of robot learning begins after

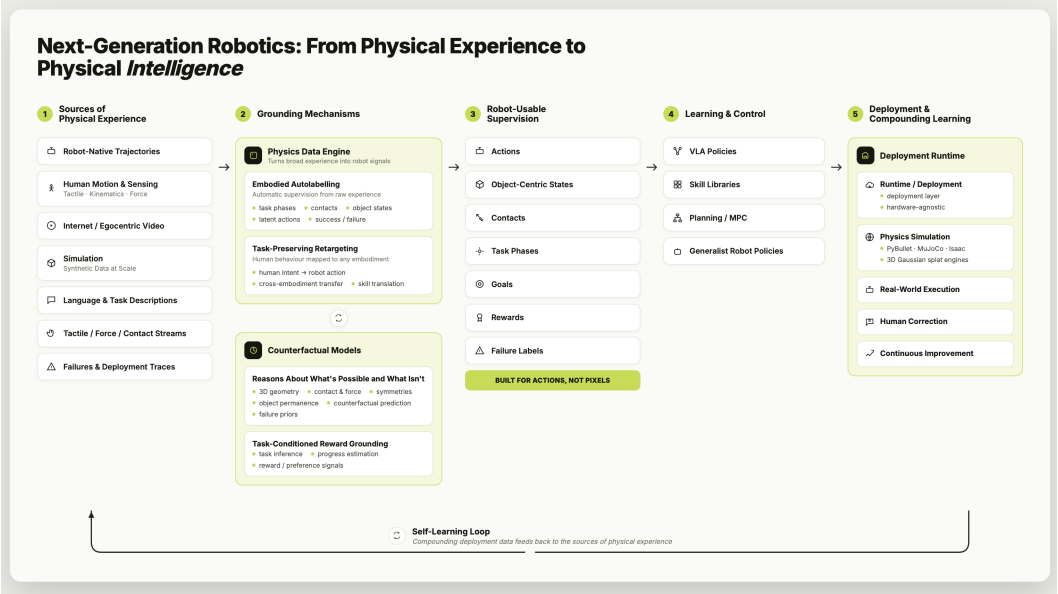

Figure 1: Next generation robotics will come from advances that go well beyond scaling vision language action (VLA) models.

data has already been made convenient for the policy: observations are paired with robot actions, demonstrations are segmented into tasks, success conditions are specified, and rewards are either hand-designed or manually labelled. This assumption is powerful, but it does not scale to the broader physical world. Human motion, internet video, wearable sensing, tactile streams, factory workflows, failures, and deployment traces all contain useful information about physical interaction, but they do not arrive as clean robot-training examples. A physical data engine is the missing layer that converts this messy experience into the variables that robot learning actually requires: object states, contacts, task phases, latent actions, goals, rewards, and success or failure labels.

The key idea is that physical experience should not be treated as raw video or unstructured logs. It should be treated as partially labelled interaction data. A person performing a task reveals more than pixels: their motion indicates intent, their hands reveal contact, object movement reveals causal structure, pauses and corrections reveal uncertainty, and the final configuration reveals something about success. Similarly, a failed robot rollout is not merely a bad trajectory; it is evidence about what the policy misunderstood, which contact was missed, which object state was unstable, or which subgoal was not achieved. Even a failure is an opportunity to learn a new skill, as long as it is properly labeled. Perhaps an object was mistakenly dropped due to a poor grasp. In a future task, dropping an object may be an essential skill. Labeling and storing detailed sub-task episodes, even failures, can build a diverse skill set to be utilized later. The role of the data engine is to recover these hidden labels automatically or semi-automatically, turning physical behaviour into supervision.

This motivates the notion of embodied autolabelling. By embodied autolabelling, we mean the process of using physical sensing, temporal structure, and world knowledge to infer robot-relevant labels from behaviour without requiring manual annotation at every step. These labels may include when a task begins and ends, which object is being manipulated, where contact occurs, what state change is intended, whether progress is being made, and whether the outcome counts as success. Unlike ordinary video labelling, embodied autolabelling is not only semantic. It must be physically grounded: it should recover labels that are useful for control, such as grasp events, force-relevant contacts, object-centric state transitions, constraints, affordances, and failure modes.

Wearable sensing makes this problem especially interesting. A motion-capture or sensorimotor suit can provide structured signals that ordinary video lacks: body pose, hand trajectories, timing, contact events, tactile cues, force-related proxies, and possibly object interaction traces. This changes the role of human demonstrations. Instead of treating a human demonstration as only a video to imitate, we can treat it as a source of physically structured supervision. The human performs the task once, but the system extracts many labels: task phase boundaries, hand-object contacts, object state changes,

intent, corrections, and candidate skill segments. These labels can then be used to train perception models, reward models, retargeting systems, world models, or robot policies. In this sense, embodied sensing is not merely a teleoperation interface; it is a labelling instrument for the physical world.

Human videos and data from wearable sensing suits also serve a dual purpose. This data can teach robots about solving tasks, but it can also teach robots about humans themselves: how humans move, use their bodies and their environment, and interact with each other. Robot intelligence should include a natural and collaborative working model of human behavior. These sources of human data should be harnessed for training human-aware, human-compliant policies, and human collaborative policies.

The central challenge is that these labels are not independent. Task phase, contact, object state, action, and reward are coupled. A contact event matters because it changes an object state; an object state matters because it defines progress toward a task; a reward matters only relative to an inferred goal; and a robot action is useful only if it preserves the task-relevant structure of the human behaviour. A physical data engine must therefore integrate perception, tracking, temporal segmentation, contact inference, language grounding, and physical reasoning into a common representation. The goal is not to build a larger dataset in the ordinary sense, but to build a system that continuously transforms physical experience into reusable robot supervision.

Raw Experiences are Heterogeneous and Asynchronous. We start from a raw episode of physical experience. This episode may come from a robot rollout, human demonstration, wearable sensing system, internet video, simulation, or deployment trace. Crucially, the different streams may not be synchronised or sampled at the same frequency. We represent those by the following:

$$\mathbf{x} = \{(v_i, \tau_i^{(v)})_{i=1}^{T_v}, (m_j, \tau_j^{(m)})_{j=1}^{T_m}, (h_k, \tau_k^{h})_{k=1}^{T_h}, (r_l, \tau_l^{(r)})_{l=1}^{T_r}, \mathtt{L}\}.$$

Here, $(v_i, \tau_i^{(v)})$ denotes video frames with timestamps, (m_j, τ_j^{m}) motion-capture, wearable, or body-pose measurements with timestamps, $(h_k, \tau_k^{(h)})$ tactile, force, contact, or hand-sensor readings with timestamps, $(r_l, \tau_l^{(r)})$ raw robot logs, if available, e.g., proprioception, deployment metadata, and $\mathtt{L}$ language associated with the episode, such as an instruction, caption, task description, or human correction.

Of course, not every episode contains all modalities. Internet video may contain only video and weak captions. A wearable-suit episode may contain video, pose, tactile, and language. A robot rollout may contain observations, actions, proprioception, and success/failure metadata. But, in its most general form, we say $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is the space of heterogeneous physical episodes.

Because the streams are asynchronous, the first hidden object is an alignment between raw observations and a common physical timeline. Let $\zeta \in \{1, \dots, Z\}$ denote a latent event timeline. This does not have to be the same as the video rate or the robot control rate. It could be continuous time, a discretised timeline, or an event-based sequence. We introduce an alignment variable: $\mathcal{A}$, such that $\mathcal{A} : \{\tau_i^{(v)}, \tau_j^{(m)}, \tau_k^{(h)}, \tau_l^{(r)}\} \to \{1, \dots, Z\}$. In words, $\mathcal{A}$ tells us which video frames, motion measurements, tactile events, robot logs, and language references correspond to the same underlying physical event. For example, it could be the case that: `video-frames:` 30-55, `motion-readings:` 102-180, and a `tactile-spike @:` 1.8 seconds, all map to the latent event of $\zeta = 2$: `contact-begins`. In this sense, temporal alignment is not a preprocessing detail. It is, in fact, part of the embodied autolabelling problem.

This immediately exposes the first interesting learning problem. Given a heterogeneous episode $\mathbf{x}$, we want to infer a sequence of latent physical events $\zeta = 1, \dots, Z$, and for each event recover the variables that would make the episode useful for robot learning. We denote the latent structure associated with event ζ by:

$$\mathbf{z}_\zeta = [\mathbf{s}_\zeta, \mathbf{c}_\zeta, \boldsymbol{\phi}_\zeta, \mathbf{u}_\zeta, \mathbf{r}_\zeta],$$

where $\mathbf{s}_\zeta$ is an object-centric physical state, $\mathbf{c}_\zeta$ is a contact or interaction label, ϕ_ζ is a task phase, $\mathbf{u}_\zeta$ is a latent physical action or transition code, and $\mathbf{r}_\zeta$ is a task-conditioned progress or reward signal. At the episode level, we also infer a goal $\mathbf{g}$ and an outcome label $\mathbf{y}$, such as success, failure, partial success, or unsafe execution. Thus, the full hidden explanation of an episode is: $\mathbf{z} = [\mathbf{z}_{1:Z}, \mathbf{g}, \mathbf{y}]$. The physical data engine can therefore be viewed as an inference model, $q_{\boldsymbol{\theta}}(\mathbf{z}, \mathcal{A}|\mathbf{x})$ which maps raw asynchronous multimodal experience into an aligned sequence of robot-relevant physical events. Importantly, $q_{\boldsymbol{\theta}}$ is not merely a perception model. It must jointly solve temporal alignment, event segmentation, object-state estimation, contact inference, phase recognition, latent-action discovery,

reward grounding, and outcome prediction. In other words, the physical data engine tries to answer: what happened, when did it happen, which objects were involved, what physical change occurred, what task was being pursued, and whether that change constituted progress or failure.

To illustrate, consider a human wearing a sensing suit while placing a cup on a tray. The raw episode may contain video frames, body-pose measurements, hand trajectories, tactile spikes, and a language instruction. The inferred event sequence might be:

$$\zeta = 1 : \texttt{reach-to-cup},\ \zeta = 2 : \texttt{contact-begins},\ \zeta = 3 : \texttt{grasp} \ldots$$

For each event, the engine should infer not only a semantic label, but a physical description: the cup pose, the hand-object contact, the relevant task phase, the latent transition being performed, and whether progress toward the goal is increasing. This is the distinction between ordinary video understanding and embodied autolabelling. A captioning model may say "a person places a cup on a tray"; a physical data engine should recover the sequence of physical events that could be retargeted, simulated, rewarded, or used to train a robot policy.

A central open question is therefore how to learn the mapping: $\mathbf{x} \rightarrow (\mathbf{z}_{1:Z}, \mathbf{g}, \mathbf{y}, \mathcal{A})$, when most episodes provide only partial supervision. Some robot episodes contain actions but weak task labels. Some videos contain task semantics but no actions. Some wearable demonstrations contain motion and contact proxies but no robot embodiment. Some simulations provide full state but imperfect realism. A useful physical data engine must combine these sources by treating them as different views of the same underlying physical structure.

3.2 Task-preserving Retargeting across Embodiments

Inferring a structured event sequence from physical experience does not by itself produce a robot policy. A human demonstration, internet video, or wearable-sensor trace may reveal what happened physically — which object moved, where contact occurred, which task phase was executed, and whether progress was made — but it still does not specify how a particular robot should act.

This is the embodiment gap. A human hand, parallel-jaw gripper, dexterous hand, mobile manipulator, quadruped, and humanoid all have different kinematics, dynamics, sensors, action spaces, contact surfaces, and failure modes. Therefore, the central question is not how to copy human motion, but how to preserve the task-relevant physical effect of that motion when executed by a different body. We call this task-preserving retargeting: the problem of mapping latent physical actions or human demonstrations into executable robot actions while preserving the intended effect on the world.

Let $\mathbf{u}_\zeta$ denote the latent physical action inferred for event ζ, and let $\mathbf{s}_\zeta$ denote the corresponding object-centric state. For a robot embodiment e, retargeting seeks an executable action or skill: $\mathbf{a}_\zeta^{(\text{embodied})} = f_\psi(\mathbf{u}_\zeta, \mathbf{s}_\zeta, \text{embodiment})$ such that the resulting robot-induced transition preserves the goal-relevant physical change $\Delta_{\mathbf{g}}(\mathrm{s}_\zeta, \mathbf{a}_\zeta^{(\text{embodied})}) \approx \Delta_{\mathbf{g}}(\mathbf{s}_\zeta, \mathbf{u}_\zeta)$. Here, $\Delta_{\mathbf{g}}$ denotes the task-relevant effect under goal $\mathbf{g}$: drawer displacement for opening, object pose for placing, relative alignment for insertion, containment for packing, or contact state for grasping. This formulation makes clear why pose matching is insufficient. The correct retargeting target is not the human joint trajectory, but the physical transformation that matters for the task.

Retargeting can preserve different invariants. At the weakest level, it preserves pose: mapping human hand or arm motion to a robot end-effector trajectory. At a stronger level, it preserves contact: ensuring that the robot touches the relevant object surfaces at the relevant moments. Stronger still, it preserves object-state transitions: ensuring that the drawer opens, the cup is lifted, or the peg becomes aligned. The strongest form preserves intent or skill: the robot may use a different motion entirely, but accomplishes the same task under the same constraints. Generalist robotics will require retargeting to move up this hierarchy, from pose-preserving imitation to task-effect-preserving translation.

This view also clarifies why wearable sensing and embodied autolabelling are valuable. A suit or sensor-rich demonstration does not need to provide the final robot action directly. Instead, it can expose the intermediate variables required for task-preserving retargeting: hand-object contact, force-relevant events, object-state changes, task phase boundaries, and latent physical actions. These variables are more transferable than raw human joint angles and more informative than video captions. They form the bridge between human physical experience and robot-executable behaviour.

3.3 Beyond Physics-Grounded World Models for Consequence Predictions

Inferring physical events and retargeting them across embodiments still leaves one central problem: a robot must reason about consequences. A candidate action is useful only if the robot can anticipate what it will do to the world. Will the object move or slip? Will contact be established or lost? Will the drawer open or jam? Will the cup remain stable after release? Will the cloth deform in the intended direction? These are not merely visual questions. They require reasoning about geometry, contact, forces, constraints, material properties, and task progress. For this reason, the next generation of robot-learning systems requires physics-grounded world models: predictive models that estimate not only what the world may look like after an action, but what physically changes and why.

The role of a physics-grounded world model is therefore different from that of a generic video generator. A video model may produce plausible future frames, but a robot needs actionable predictions. It must know whether an action produces the desired object-state transition, whether a grasp is stable, whether a collision will occur, whether an insertion will fail because of misalignment, or whether an object will fall after being released. Thus, a robot world model should operate over structured physical variables whenever possible: object poses, spatial relations, contacts, constraints, velocities, forces, deformable states, and physical properties such as friction, mass, stiffness, or compliance. These variables are precisely the quantities that determine whether an imagined action can become a successful real-world behaviour.

We can write this role abstractly as consequence prediction. Given an object-centric state $\mathbf{s}_\zeta$, a goal $\mathbf{g}$, and either a latent physical action $\mathbf{u}_\zeta$, or an embodiment-specific robot action $\mathbf{a}_\zeta^{(\text{embodied})}$, the world model predicts the next physical state:

$$\mathbf{s}_{\zeta+1} \sim p_\omega(\cdot|\mathbf{s}_\zeta, \mathbf{u}_\zeta, \mathbf{g}).$$

For a specific robot embodiment, this would amount to:

$$\mathbf{s}_{\zeta+1} \sim p_\omega(\cdot|\mathbf{s}_\zeta, \mathbf{a}_\zeta^{(\text{embodied})}, \text{embodiement}, \mathbf{g}).$$

The first form supports task-level reasoning: what physical transition should occur if the intended action is "pull", "lift", "insert", or "place"? The second supports embodiment-specific planning: what will happen if this particular robot, with this morphology and controller, executes this action from this state? In both cases, the model should predict more than pixels. It should predict the physical variables that matter for control and reward.

This gives the world model a central role in the proposed stack. It can be used before action execution to evaluate candidate retargeted actions, during planning to search over alternatives, after failures to explain what went wrong, and during training to generate counterfactual experience. For example, if a human demonstration suggests the latent action "pull drawer outward", a retargeting model may propose several robot motions. A physics-grounded world model can evaluate which motion is likely to establish the correct contact, apply force along the right direction, avoid collision, and produce the desired drawer displacement. Similarly, if a robot fails to insert a peg, the world model can help distinguish whether the failure came from poor alignment, insufficient force, unstable grasp, object geometry, or a wrong task phase.

The most important point is that consequence prediction should be task-conditioned. A world model does not need to predict every detail of the future equally well. It needs to predict the aspects of the future that are relevant to the task. For opening a drawer, drawer displacement and handle contact matter more than the exact background texture. For pouring, liquid state and container pose matter more than the appearance of the table. For folding cloth, deformable geometry and contact points matter more than pixel-perfect video reconstruction. This suggests that the objective of a robot world model should be aligned with downstream control, not only with visual reconstruction. The question is not "does the future look realistic?" but "does the prediction preserve the physical consequences that determine success or failure?"

This also clarifies why physics grounding matters. Purely learned predictors can exploit visual regularities without understanding the underlying constraints. They may generate futures in which objects interpenetrate, contacts occur without force, rigid objects deform unrealistically, or effects appear without plausible causes. Physics-grounded models reduce these failure modes by injecting structure: object permanence, geometric consistency, conservation laws, differentiable contact, learned or explicit material parameters, action-conditioned dynamics, and uncertainty estimates. The

goal is not necessarily to build a perfect simulator. Rather, it is to learn a predictive model that is accurate where control needs accuracy and uncertain where the system lacks evidence.

In this view, world models are not separate from embodied autolabelling and retargeting. They complete the loop. The physical data engine infers what happened; retargeting proposes how a robot could reproduce the relevant physical effect; the world model predicts what would happen if the robot tried. These three components can improve one another. Autolabelled contact and object-state transitions provide supervision for the world model. The world model can detect inconsistent labels or impossible transitions. Retargeting can use world-model rollouts to choose actions that preserve task effects. Deployment failures can then be fed back into the data engine as new examples of what the model failed to predict.

The open challenge is to decide what representation such a world model should use. Pixel-space prediction offers broad coverage but weak physical abstraction. Object-centric models expose entities and relations but require reliable perception and tracking. 3D representations such as point clouds, meshes, neural fields, or Gaussian splats provide geometry but may still struggle with contact, force, and material response. Mechanics-based models encode physical laws but can be brittle when the environment is unknown or deformable. The most promising direction may be hybrid: learned models that combine 3D scene representations, object-centric structure, physics-inspired constraints, and data-driven residual dynamics. Such models would not merely imagine the future; they would provide physically meaningful counterfactuals for robot learning.

Physics-grounded world models therefore address a missing component in the path from physical experience to physical intelligence. They turn data into counterfactuals. Without them, a robot can only imitate what it has seen or execute what it has been commanded. With them, it can ask what would happen under alternative actions, bodies, contacts, and goals. But this promise depends on grounding: imagined futures are useful for robotics only when they preserve the physical structure that makes actions succeed or fail.

3.4 Self-Improving Deployment Loops

After a robot executes an action, the central question is no longer only what happened, but whether what happened was useful. A world model may predict that a cup will move, a drawer will open, or an object will fall; a retargeting system may produce a physically feasible action; and a policy may execute that action in the real world. But learning from the result requires a task-conditioned interpretation of the outcome. Did the action make progress? Did it solve the intended task? Did it fail because of perception, contact, force, timing, planning, or embodiment mismatch? Was the final state good or bad relative to the goal? These questions cannot be answered by a generic state evaluator. They require reward grounding: the ability to assign progress, success, and failure relative to the task being attempted.

This is why robotic reward models should be task-conditioned. A physical state is not intrinsically successful or unsuccessful. The same state can mean different things depending on the goal: a cup resting on a table is success for "put the cup down", failure for "pick up the cup", and irrelevant for "open the drawer". We can express this by writing reward as: $\mathbf{r}_\eta(\mathbf{s}_\zeta, \mathbf{g}, \phi_\zeta)$, where $\mathbf{s}_\zeta$ is the inferred physical state at event ζ, $\mathbf{g}$ is the task or goal, and ϕ_ζ is the task phase. In this view, reward is not merely a scalar attached to a state. It is an interpretation of physical progress under a goal. A good reward model should estimate whether the relevant contact occurred, whether the object moved in the intended way, whether the system entered a recoverable or unrecoverable failure mode, and whether the final configuration satisfies the task.

This perspective also connects reward learning to video and language understanding. Many tasks can be recognised from the temporal structure of a demonstration or rollout: approaching, contacting, manipulating, releasing, verifying, recovering. Videos often reveal not only what task is being attempted, but also what progress and failure look like. Language can provide the task hypothesis; video and state estimates can provide evidence; human feedback can resolve ambiguity. Thus, reward grounding can be seen as task-conditioned physical interpretation: given a goal and an observed trajectory, infer which events count as progress, which count as failure, and which final states count as success. This is different from generic preference modelling. The reward must be tied to object states, contacts, constraints, and task phases.

This reward-grounding problem is what makes self-improving deployment possible. In a deployed robot system, every rollout should become more than a pass/fail record. It should become a labelled physical episode. A successful rollout provides examples of robust task completion. A failed rollout provides information about missing contact, wrong object state, unstable grasp, poor alignment, unsafe motion, or reward misinterpretation. A human correction provides a high-value supervision signal: it reveals not only that the robot was wrong, but often how the task should have proceeded. If these outcomes are fed back into the physical data engine, the system can update its reward model, retargeting model, world model, and policy.

The resulting loop is: deploy policy $\rightarrow$ observe outcome $\rightarrow$ infer task-conditioned progress / success / failure $\rightarrow$ explain failure or correction $\rightarrow$ add grounded supervision to the data engine $\rightarrow$ update reward model, world model, retargeting, and policy $\rightarrow$ redeploy. This is the key difference between a robot that merely executes a trained policy and a robot-learning system that compounds over time. Without reward grounding, deployment traces are difficult to use: a failure is just a failed video, and a success is just an episode that happened to work. With task-conditioned reward grounding, deployment traces become structured supervision. The system can ask: which subgoal failed, which contact was missing, which object state was wrong, and what alternative action would have improved the outcome?

A self-improving deployment loop therefore requires three capabilities. First, it must monitor execution and detect meaningful events: contacts, state changes, subgoal completion, anomalies, and safety violations. Second, it must evaluate those events relative to the task, producing progress, reward, and failure labels. Third, it must route the resulting supervision to the right component: policy updates when the action was poor, world-model updates when the consequence prediction was wrong, retargeting updates when the physical effect was not preserved, and reward-model updates when success or failure was misclassified. This component-level credit assignment is essential. Otherwise, the system may know that a rollout failed but not what should change.

This closes the pipeline introduced in the paper. A physical data engine turns heterogeneous experience into latent physical events. Task-preserving retargeting maps those events to robot actions. Physics-grounded world models predict the consequences of those actions. Task-conditioned reward grounding interprets the outcomes. Deployment then supplies new episodes that re-enter the same pipeline. The long-term goal is a compounding physical-intelligence system: one in which every demonstration, video, simulation rollout, robot failure, and human correction becomes structured supervision for the next generation of robot behaviour.

4 Conclusions

This position paper has argued that the next stage of generalist robotics should not be framed only as a policy-scaling problem. Vision-language-action models, large robot datasets, simulation pipelines, and learned world models are all important parts of the emerging robotics stack. However, they do not by themselves solve the central bottleneck: how to convert the world's broad, messy, and weakly labelled physical experience into supervision that a robot can actually use.

The key limitation is not simply a shortage of data. The world already contains enormous amounts of physical behaviour: humans manipulating objects, tools being used, factories operating, homes being organised, robots succeeding and failing, and simulations generating counterfactual interactions. The difficulty is that this experience does not arrive in the coordinate system of robot learning. It usually lacks robot actions, embodiment-specific constraints, task-phase labels, contact annotations, reward signals, and success or failure explanations. The missing problem is therefore grounding: transforming physical experience into robot-usable variables such as object states, contacts, latent actions, task phases, goals, rewards, and physically meaningful counterfactuals.

This perspective suggests that VLAs should be understood as one layer in a larger physical-intelligence stack. They provide a powerful policy interface between perception, language, and action, but their effectiveness depends on upstream and downstream mechanisms that make physical experience learnable. A robot-learning system must be able to autolabel heterogeneous behaviour, retarget task-relevant physical effects across embodiments, predict the consequences of candidate actions, and interpret deployment outcomes relative to the task being attempted. Without these components, larger policies may improve performance on curated robot-native datasets, but they will remain limited by the amount of experience that has already been manually or implicitly grounded for them.

This also suggests a different set of evaluation questions for generalist robotics. Instead of asking only whether a larger policy solves more tasks, we should ask whether a system can convert weaker sources of physical experience into useful supervision. Can it infer contacts, object-state changes, and task phases from human behaviour? Can it retarget a demonstrated physical effect to a new embodiment without merely copying pose? Can its world model predict the consequences that matter for success and failure, rather than only generating plausible future frames? Can its reward model distinguish progress, failure, recovery, and success relative to the current goal? Can deployment failures update the right component of the stack: the policy, the reward model, the world model, or the retargeting mechanism? These questions define the grounding agenda for robotics beyond VLA scaling.

The broader implication is that the next foundation model for robotics may not be a single monolithic model. It may instead be a compounding system: a physical data engine that turns heterogeneous experience into structured supervision; an embodiment interface that maps task-relevant effects to robot actions; a physics-grounded world model that generates actionable counterfactuals; and a task-conditioned deployment loop that converts successes, failures, and corrections into future improvement. In such a system, every human demonstration, internet video, simulation rollout, tactile trace, robot failure, and human correction becomes part of a growing supervision engine for physical intelligence.

Robots, therefore, need more than VLAs. They need the architectural pillars that make physical experience usable. Progress in robotics will depend not only on scaling policies but on building the grounding mechanisms that connect the world's behavioural data to robot actions, rewards, models, and continual deployment. The central challenge for the field is to move from robot-native datasets to world-scale physical supervision, and from isolated policies to systems that learn from the physical world itself.

References

Jad Abou-Chakra, Lingfeng Sun, Krishan Rana, Brandon May, Karl Schmeckpeper, Niko Suenderhauf, Maria Vittoria Minniti, and Laura Herlant. Real-is-sim: Bridging the sim-to-real gap with a dynamic digital twin, 2025. URL `https://arxiv.org/abs/2504.03597`.

Maximilian Adang, JunEn Low, Ola Shorinwa, and Mac Schwager. Singer: An onboard generalist vision-language navigation policy for drones. In *In Proc. of the International Conference on Robotics and Automation (ICRA)*, 2026.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL `https://arxiv.org/abs/2506.09985`.

Tewodros W Ayalew, Xiao Zhang, Kevin Yuanbo Wu, Tianchong Jiang, Michael Maire, and Matthew R Walter. Progressor: A perceptually guided reward estimator with self-supervised online refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10297–10306, 2025.

Shuanghao Bai, Meng Li, Xinyuan Lv, Jiawei Wang, Xinhua Wang, Fei Liao, Chengkai Hou, Langzhe Gu, Wanqi Zhou, Kun Wu, et al. Hex: Humanoid-aligned experts for cross-embodiment whole-body manipulation. *arXiv preprint arXiv:2604.07993*, 2026.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016. URL http://arxiv.org/abs/1612.00222.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=bJbSbJskOS.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics, 2017. URL https://arxiv.org/abs/1612.00341.

Annie S. Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos, 2021. URL https://arxiv.org/abs/2103.16817.

Qianzhong Chen, Naixiang Gao, Suning Huang, JunEn Low, Timothy Chen, Jiankai Sun, and Mac Schwager. Grad-nav++: Vision-language model enabled visual drone navigation with gaussian radiance fields and differentiable dynamics. *IEEE Robotics and Automation Letters*, 11(2):1418–1425, 2025a.

Qianzhong Chen, Jiankai Sun, Naixiang Gao, JunEn Low, Timothy Chen, and Mac Schwager. Grad-nav: Efficiently learning visual drone navigation with gaussian radiance fields and differentiable dynamics. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7941–7948. IEEE, 2025b.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Yide Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Alexander I Cowen-Rivers, Daniel Palenicek, Vincent Moens, Mohammed Amin Abdullah, Aivar Sootla, Jun Wang, and Haitham Bou-Ammar. Samba: Safe model-based & active reinforcement learning. *Machine Learning*, 111(1):173–203, 2022.

Miles D. Cranmer, Sam Greydanus, Stephan Hoyer, Peter W. Battaglia, David N. Spergel, and Shirley Ho. Lagrangian neural networks. *CoRR*, abs/2003.04630, 2020. URL https://arxiv.org/abs/2003.04630.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *CoRR*, abs/1910.11215, 2019. URL http://arxiv.org/abs/1910.11215.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, et al. Actioncodec: What makes for good action tokenizers. *arXiv preprint arXiv:2602.15397*, 2026.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Chrisantus Eze and Christopher Crick. Learning by watching: A review of video-based learning approaches for robot manipulation, 2025. URL https://arxiv.org/abs/2402.07127.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware vla pretraining through visual-physical alignment from human videos. *arXiv preprint arXiv:2512.13080*, 2025.

Zhiyuan Feng, Qixiu Li, Huizhi Liang, Rushuai Yang, Yichao Shen, Zhiying Du, Zhaowei Zhang, Yu Deng, Li Zhao, Hao Zhao, et al. From human videos to robot manipulation: A survey on scalable vision-language-action learning with human-centric data. 2026.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: Object-centric world models for robotics manipulation, 2023. URL https://arxiv.org/abs/2307.02427.

Figure AI. Helix: A vision-language-action model for generalist humanoid control. https://www.figure.ai/news/helix, February 2025. Accessed: 2026-04-28.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Abraham George and Amir Barati Farimani. One act play: Single demonstration behavior cloning with action chunking transformers. *arXiv preprint arXiv:2309.10175*, 2023.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *CoRR*, abs/1906.01563, 2019. URL http://arxiv.org/abs/1906.01563.

David Ha and Jürgen Schmidhuber. World models. 2018a. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution, 2018b. URL https://arxiv.org/abs/1809.01999.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, pages 2555–2565, 2019a. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *CoRR*, abs/1912.01603, 2019b. URL http://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *CoRR*, abs/2010.02193, 2020. URL https://arxiv.org/abs/2010.02193.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Helong Huang, Min Cen, Kai Tan, Xingyue Quan, Guowei Huang, and Hong Zhang. Graphcot-vla: A 3d spatial-aware reasoning vision-language-action model for robotic manipulation with ambiguous instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18324–18332, 2026a.

Suning Huang, Qianzhong Chen, Xiaohan Zhang, Jiankai Sun, and Mac Schwager. Particleformer: A 3d point cloud world model for multi-object, multi-material robotic manipulation, 2025. URL https://arxiv.org/abs/2506.23126.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation, 2026b. URL https://arxiv.org/abs/2601.03782.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), January 2019. ISSN 2470-9476. doi: 10.1126/scirobotics.aau5872. URL http://dx.doi.org/10.1126/scirobotics.aau5872.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment, 2019. URL https://arxiv.org/abs/1909.12271.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: zero-shot task generalization with robotic imitation learning. *CoRR*, abs/2202.02005, 2022. URL https://arxiv.org/abs/2202.02005.

Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation, 2025. URL https://arxiv.org/abs/2503.06170.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, et al. Wholebodyvla: Towards unified latent vla for whole-body loco-manipulation control. *arXiv preprint arXiv:2512.11047*, 2025.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. 2023.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11):1238–1274, 2013. doi: 10.1177/0278364913495721. URL http://sagepub.com.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems (RSS)*, Virtual Conference, 2021.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Chenhao Li, Andreas Krause, and Marco Hutter. Uncertainty-aware robotic world model makes offline model-based reinforcement learning work on real robots. *arXiv preprint arXiv:2504.16680*, 2025a.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2026.

Xiaoqi Li, Liang Heng, Jiaming Liu, Yan Shen, Chenyang Gu, Zhuoyang Liu, Hao Chen, Nuowei Han, Renrui Zhang, Hao Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025b.

Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator, 2025c. URL `https://arxiv.org/abs/2411.11839`.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL `https://arxiv.org/abs/2306.03310`.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110, 2024a.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024b.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, et al. Faster: Toward efficient autoregressive vision language action modeling via neural action tokenization. *arXiv preprint arXiv:2512.04952*, 2025a.

Yuyang Liu, Chuan Wen, Yihang Hu, Dinesh Jayaraman, and Yang Gao. Timerewarder: Learning dense reward from passive videos via frame-wise temporal distance. *arXiv preprint arXiv:2509.26627*, 2025b.

JunEn Low, Maximilian Adang, Javier Yu, Keiko Nagami, and Mac Schwager. Sous vide: Cooking visual drone navigation policies in a gaussian splatting vacuum. *IEEE Robotics and Automation Letters*, 2025.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. *arXiv preprint arXiv:2508.17600*, 2025.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *CoRR*, abs/1907.04490, 2019. URL `http://arxiv.org/abs/1907.04490`.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36: 655–677, 2023.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023. URL `https://arxiv.org/abs/2310.17596`.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022. URL https://arxiv.org/abs/2112.03227.

Zhiting Mei, Tenny Yin, Micah Baker, Ola Shorinwa, and Anirudha Majumdar. World models that know when they don't know: Controllable video generation with calibrated uncertainty. *arXiv preprint arXiv:2512.05927*, 2025.

Perusha Moodley, Pramod Kaushik, Dhillu Thambi, Mark Trovinger, Praveen Paruchuri, Xia Hong, and Benjamin Rosman. Multi-state-action tokenisation in decision transformers for multi-discrete action spaces. *arXiv preprint arXiv:2407.01310*, 2024.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations, 2021. URL https://arxiv.org/abs/2107.14483.

Fabio Muratore, Fabio Ramos, Greg Turk, Wenhao Yu, Michael Gienger, and Jan Peters. Robot learning from randomized simulations: A review, 2022. URL https://arxiv.org/abs/2111.00956.

Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, and Yang Gao. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations, 2026. URL https://arxiv.org/abs/2602.06643.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026.

Authors of PIN-WM. Pin-wm: Learning physics-informed world models for non-prehensile manipulation. *arXiv preprint arXiv:2504.16693*, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

Maxim A Patratskiy, Alexey K Kovalev, and Aleksandr I Panov. Spatial traces: Enhancing vla models with spatial-temporal understanding. *Optical Memory and Neural Networks*, 34(Suppl 1):S72–S82, 2025.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Kyriakos Polymenakos, Alessandro Abate, and Stephen Roberts. Safe policy search using gaussian process models. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems*, pages 1565–1573, 2019.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025a. URL https://arxiv.org/abs/2501.15830.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025b.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control, 2018. URL https://arxiv.org/abs/1806.01242.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks, 2020. URL https://arxiv.org/abs/2002.09405.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video, 2018. URL https://arxiv.org/abs/1704.06888.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

Skild AI. Building the general-purpose robotic brain. https://www.skild.ai/blogs/building-the-general-purpose-robotic-brain, July 2025. Accessed: 2026-04-28.

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos, 2020. URL https://arxiv.org/abs/1912.04443.

Harris Song and Long Le. Avi: Action from volumetric inference. *arXiv preprint arXiv:2510.21746*, 2025.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation, 2024. URL https://arxiv.org/abs/2403.03949.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

Maggie Wang, Stephen Tian, Aiden Swann, Ola Shorinwa, Jiajun Wu, and Mac Schwager. Phys2real: Fusing vlm priors with interactive online adaptation for uncertainty-aware sim-to-real manipulation. In *In Proc. of the International Conference on Robotics and Automation (ICRA)*, 2026a.

Meizhong Wang, Wanxin Jin, Kun Cao, Lihua Xie, and Yiguang Hong. Contactgaussian-wm: Learning physics-grounded world model from videos, 2026b. URL https://arxiv.org/abs/2602.11021.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024. URL https://arxiv.org/abs/2311.01455.

Isaac R Ward, Michelle Ho, Houjun Liu, Aaron Feldman, Joseph Vincent, Liam Kruse, Sean Cheong, Duncan Eddy, Mykel J Kochenderfer, and Mac Schwager. Foundational world models accurately detect bimanual manipulator failures. In *In Proc. of the International Conference on Robotics and Automation (ICRA)*, 2026.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning, 2022. URL https://arxiv.org/abs/2206.14176.

Yuxuan Wu, Lei Pan, Wenhua Wu, Guangming Wang, Yanzi Miao, Fan Xu, and Hesheng Wang. Rl-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning, 2025. URL https://arxiv.org/abs/2409.20291.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics, 2024. URL https://arxiv.org/abs/2311.12198.

Haoru Xue, Xiaoyu Huang, Dantong Niu, Qiayuan Liao, Thomas Kragerud, Jan Tommy Gravdahl, Xue Bin Peng, Guanya Shi, Trevor Darrell, Koushil Sreenath, et al. Leverb: Humanoid whole-body control with latent vision-language instruction. *arXiv preprint arXiv:2506.13751*, 2025.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators, 2024a. URL https://arxiv.org/abs/2310.06114.

Yanting Yang, Minghao Chen, Qibo Qiu, Jiahao Wu, Wenxiao Wang, Binbin Lin, Ziyu Guan, and Xiaofei He. Adapt2reward: Adapting video-language models to generalizable robotic rewards via failure prompts. In *European Conference on Computer Vision*, pages 163–180. Springer, 2024b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Naveed Malik, Kyungmin Lee, William Liang, Nadun Ranawaka Arachchige, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Danfei Xu, Yilun Du, Ryan Julian, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026. URL https://openreview.net/forum?id=cd33uUB609.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning, 2021. URL https://arxiv.org/abs/2106.03911.

Haochen Zhang, Nader Zantout, Pujith Kachana, Zongyuan Wu, Ji Zhang, and Wenshan Wang. Vla-3d: A dataset for 3d semantic scene understanding and navigation. *arXiv preprint arXiv:2411.03540*, 2024.

Jiafan Zhang. Act-tsa: Action chunking transformer with two-stage attention for temporal multimodality in bimanual manipulation tasks. In *2025 IEEE 7th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 764–772. IEEE, 2025.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. Rewind: Learning new tasks from language without new demonstrations. In *Inductive Biases in Reinforcement Learning Workshop@ RLC 2025*, 2025a.

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions, 2025b. URL https://arxiv.org/abs/2511.04665.

Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *CoRR*, abs/1909.12077, 2019. URL http://arxiv.org/abs/1909.12077.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024. URL https://arxiv.org/abs/2404.12377.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.