A Review of Learning-Based Dynamics Models for Robotic Manipulation

Bo Ai[1*], Stephen Tian[2], Haochen Shi[2], Yixuan Wang[3], Tobias Pfaff[4], Cheston Tan[5], Henrik I. Christensen[1], Hao Su[1,6], Jiajun Wu[2], Yunzhu Li[3*]

[1]University of California San Diego, USA

[2]Stanford University, USA

[3]Columbia University, USA

[4]Google DeepMind, UK

[5]Agency for Science, Technology and Research, Singapore

[6]Hillbot, USA

[*]Corresponding authors: Bo Ai ⟨bai@ucsd.edu⟩ and Yunzhu Li ⟨yunzhu.li@columbia.edu⟩.

Dynamics models that predict the effects of physical interactions are essential for planning and control in robotic manipulation. Although models based on physical principles often generalize well, they typically require full-state information, which can be difficult or impossible to extract from perception data in complex, real-world scenarios. Learning-based dynamics models provide an alternative by deriving state transition functions purely from perceived interaction data, enabling the capture of complex, hard-to-model factors, predictive uncertainty, and accelerating simulations that are often too slow for real-time control. Recent successes in this field have demonstrated notable ad-

vancements in robot capabilities, including long-horizon manipulation of deformable objects, granular materials, and complex multi-object interactions like stowing and packing. A crucial aspect of these investigations is the choice of state representation, which determines the inductive biases in the learning system for reduced-order modeling of scene dynamics. This review provides a timely and comprehensive review of current techniques and trade-offs in designing learned dynamics models, highlighting their role in advancing robot capabilities through integration with state estimation and control, and identifying critical research gaps for future exploration.

Summary: Learning dynamics models across diverse representations from robot-world interactions improves robotic manipulation.

Introduction

Humans possess an intuitive grasp of physics that lets us interact with the environment and predict its evolution (*1*). By processing multisensory information, we form mental models that help us anticipate how our actions affect the world (*2*) (Figure 1). This intuitive understanding of physics does not depend on analytical methods. Yet, it applies across materials and objects, supporting diverse interactive skills that far exceed those of current robots.

Emulating this intuitive physics understanding in robots equates to deriving predictive models that anticipate action outcomes and support effective planning. Physics-based models (*3, 4*) generalize well but rely on full-state information, which is often unattainable in real-world manipulation tasks. Learning-based models offer an alternative by learning predictive dynamics directly from raw sensory data, capturing hard-to-model factors (*5, 6*), reasoning about uncertainty (*6–8*), and accelerating high-precision simulations too slow for real-time control (*9, 10*).

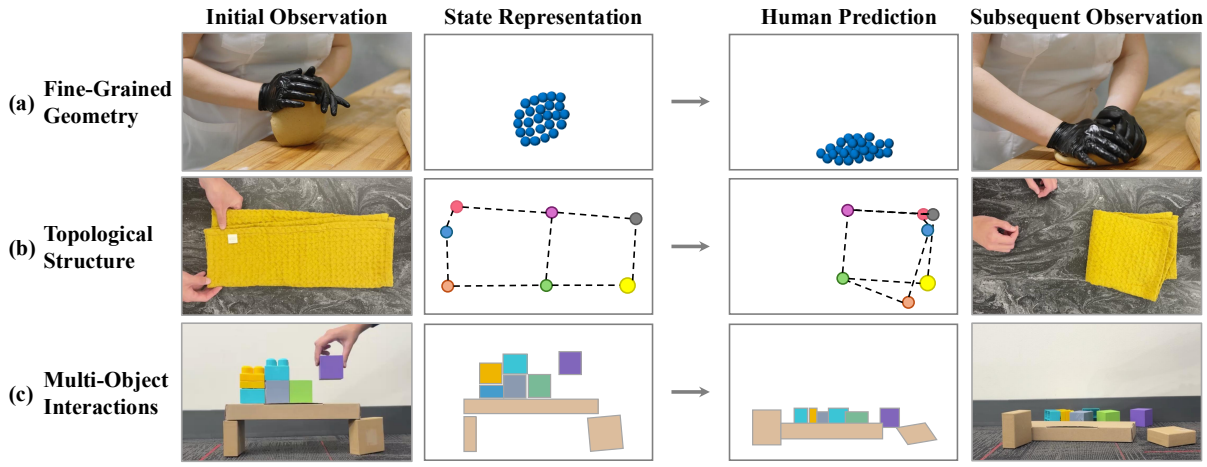

Figure 1: **Human intuitive physics for manipulation tasks.** Humans rely on intuitive physics to perform complex manipulation tasks. Depending on the task and environment, different levels of abstraction may be used in mental representations: (a) particles for fine-grained geometry, (b) keypoints for structural details, and (c) object-centric representations for multi-object interactions. We illustrate mental predictions of manipulation actions based on these abstractions. In robotics, learning-based dynamics models aim to equip robots with similar predictive capabilities using structured state representations.

Recent advances leverage deep neural networks as function approximators (*8, 11*).

Despite their promise, learning-based dynamics models face a fundamental challenge: designing inductive biases that ensure sample efficiency and generalization (*12*). This is particularly critical in robotics, where real-world data collection is costly and open-world environments have vast state spaces. Effective models require compact state representations and structured priors to efficiently process this information. However, this introduces trade-offs—although compact state spaces enhance generalization, they may reduce model expressiveness or complicate state estimation. Addressing these challenges requires careful consideration of task requirements, environmental complexity, and sensory modalities.

This review provides a comprehensive analysis of learning-based dynamics models, examining trade-offs in state representations and model architectures, as well as their effects on robotic capabilities. We discuss perception requirements for state estimation, model architectures for

learning state transitions, and how different representations influence sample efficiency, generalization, and task suitability. Given increasing integration of learning-based dynamics models with planning for manipulation—spanning object repositioning (*13–15*), deformable object handling (*16–19*), multi-modal perception (*20, 21*), and multi-object interaction (*14, 22, 23*)—a discussion on model design and implications for planning is crucial. Although prior reviews focused on related topics such as deformable object manipulation (*24, 25*), physics-based simulation (*26,27*), and intuitive physics (*28*), a dedicated review of learning-based dynamics models is lacking. This work fills that gap, providing insights for future research in robotic manipulation and beyond.

This review focuses on the intersection of learning-based dynamics models and robotic manipulation. Thus, analytical dynamics models (e.g., (*29*)), differentiable (but not learned) models (e.g., (*27*)), and hybrid models (e.g., (*5, 6*)) are beyond its scope. Similarly, learning-based dynamics models without demonstrated applications to robotic manipulation are not covered comprehensively. Within this scope, we begin by introducing learning-based dynamics models and contrasting them with analytical simulators. We then present a taxonomy of models based on state representations, discussing associated perception and dynamics learning techniques. Subsequently, we explore how planning algorithms and policy learning can integrate these learned dynamics models to enable robotic capabilities. We end with discussions on future directions and challenges in the field.

Learning-Based Dynamics Models

Learning-based dynamics models predict how the world evolves in response to actions. This article focuses on models of environment dynamics external to the robot.

Background

We use the framework of Partially Observable Markov Decision Processes (POMDPs) to formalize the process of perceiving and acting.

At time t, the agent is in state $s_t \in \mathcal{S}$, where $\mathcal{S}$ denotes the state space. It receives an observation $o_t \in \mathcal{O}$ from the environment, and then takes an action a_t based on a policy π, i.e., $a_t = \pi(o_t)$ and $a \in \mathcal{A}$. Conditioned on this action, the environment transits to the next state $s_{t+1} = \mathcal{T}(s_t, a_t)$, where $\mathcal{T}$ is the environment transition function. The process repeats until the task objective is achieved or the number of time steps reaches the task horizon H. The agent's goal is to find a policy that minimizes the cost function c defined on the state s_t and action a_t over the time horizon, defined as

$$\min_{\pi} \ \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{H} c(s_t, a_t) \right]. \tag{1}$$

This formulation underlies both classical robotic control and model-based reinforcement learning: in both paradigms, a transition model simulates future trajectories for planning or policy learning. The key question is how to represent and construct the policy function π. Learned-model based approaches approximate the transition model using a learned function $\hat{\mathcal{T}}$ before using it for control. We examine the core components of this framework below:

Perception: The perception module g estimates the environment state s_t from past observations $o_{0:t}$ and actions $a_{0:t-1}$, i.e., $s_t = g(o_{0:t}, a_{0:t-1})$, which simplifies to $s_t = g(o_t, a_{t-1})$ in fully observable settings. We view s_t as a unified representation of all task-relevant information inferred from raw sensory data, serving as input to downstream processes. A central challenge lies in defining s_t to capture minimal yet sufficient information for manipulation. This review surveys different choices of s_t and their trade-offs.

Dynamics: The dynamics model $\hat{\mathcal{T}}$ predicts the state transition from s_t to s_{t+1} given action a_t. Its design is closely coupled with the structure of s_t, often leveraging inductive biases to improve generalization and data efficiency. For instance, graph neural networks (GNNs) naturally suit particle-based states due to their spatial equivariance. This review examines model architectures for $\hat{\mathcal{T}}$ across different state representations.

Control: The policy π generates control signals to minimize the cost (Equation 1). It can be implemented via planning or policy learning, and can output position-based (e.g., end-effector poses) or force-based control signals (e.g., joint torques). Its design directly impacts computational efficiency and control quality. This review examines how control algorithms integrate with dynamics models towards solving concrete manipulation tasks.

Figure 2 illustrates how dynamics models are learned from physical interaction data and integrated with control for downstream tasks.

Comparison with Physics-Based Dynamics Models

Traditionally, dynamics models are defined by analytical solvers that integrate governing equations of motion. In robotics, this includes rigid-body simulators (*30, 31*) and deformable object solvers such as the material point method (MPM) (*4*). Despite decades of progress, these models often diverge from real-world behavior—a challenge known as the *sim-to-real gap* (*32*). This gap arises because many physical effects—such as frictional contact or actuator drift—are difficult to model precisely or require parameters that are hard to measure (*33*). Even with accurate parameters, missing latent factors like temperature can undermine accuracy. Moreover, real-world deployment depends on accurate state estimation and system identification, where small errors can accumulate over time (*34*).

Learned dynamics models offer an alternative by directly capturing physical processes from

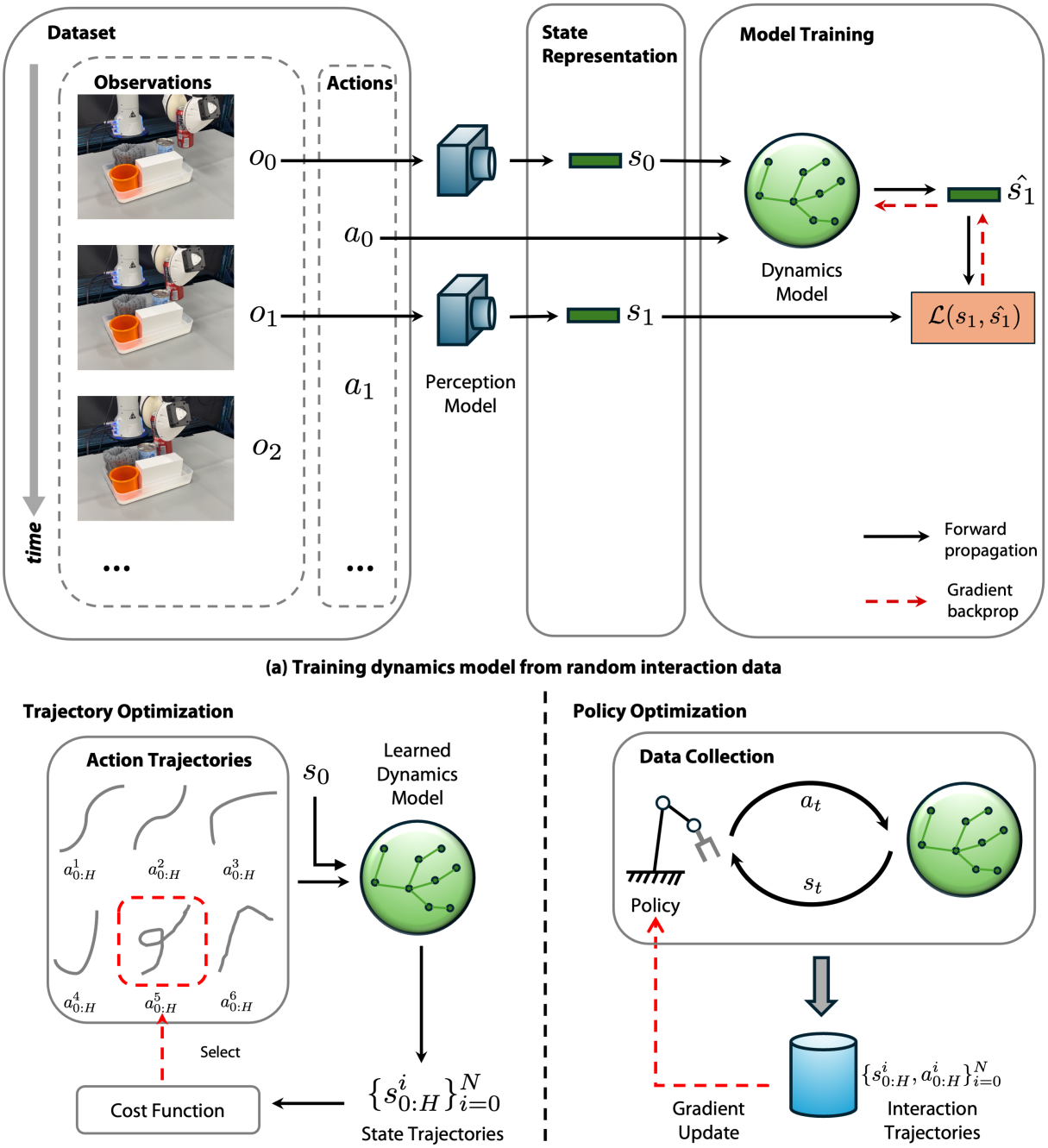

Figure 2: **Robotic manipulation using learning-based dynamics models.** (a) The dynamics model is trained on interaction data. The perception model extracts state representations s_t from observations o_t. Dynamics are learned in a self-supervised fashion. (b) The learned dynamics model is applied for downstream control, either by evaluating action trajectories $\{a^i_{0:H}\}^N_{i=0}$ for planning or by generating interaction data $\{s^i_{0:H}, a^i_{0:H}\}^N_{i=0}$ for policy learning.

interaction data, mitigating the sim-to-real gap. They can compensate for state estimation errors (*35*) or bypass state estimation entirely when trained on raw sensory inputs. Beyond bridging this gap, learned models are also end-to-end differentiable, enabling gradient-based planning, control, and online adaptation. Some studies find that learned models offer smoother gradients than analytical solvers (*36*), and can be more computationally efficient, especially for non-rigid systems (*37*).

State Representations

How should we represent the state of the world for learning-based dynamics models? A natural starting point is raw observations, such as **pixels** encoding RGB, depth, or force fields. However, the state needs only capture aspects of the environment relevant for accurate prediction and decision-making. This motivates compact and structured representations.

One approach is to use **latent representations**, which compress raw observations into lower-dimensional encodings but often lack explicit 3D structure. To incorporate geometry, the world can be discretized into **particles**, representing surfaces and volumes in 3D. For many tasks, particles may be overly detailed, and **keypoints** offer a more abstract alternative by capturing salient task-relevant features. Yet, these representations often treat the scene as unstructured collections of elements, whereas humans perceive and interact with discrete entities. **Object-centric** representations explicitly model objects and their interactions, adding structure beyond lower-level elements.

These representations reflect varying levels of abstraction and modeling assumptions. More abstract representations enable reduced-order modeling, improved sample efficiency, and generalization by focusing on task-relevant dynamics but often require stronger perception priors, such as object segmentation or keypoint detection. This section surveys perception methods and dynamics models for each representation and discusses trade-offs and practical considerations.

Pixel-Based Representations

The most straightforward state representation is observed raw pixels (e.g., two-dimensional feature maps of RGB(D) data). They are typically obtained directly from cameras and usually require only simple post-processing such as downsampling or cropping.

Dynamics Learning

Learning dynamics models in pixel space can be framed as *action-conditioned video prediction*. Given one or more context frames) $o_{0:t}$ and a sequence of actions $a_{0:H}$, the model predicts future observations $o_{t+1:H}$ conditioned on the agent taking those actions. Although video prediction and generation are widely studied in computer vision and employ similar techniques as pixel dynamics models, we focus on models that focus on physical prediction and action-conditioned models for planning.

An early line of work applying pixel space models to planning with physical robots is visual foresight (*38*). This method trained a flow-based action-conditioned video dynamics model on robotic physical interaction data and used it to plan object-pushing tasks. Extensions of this work demonstrated application to robotic tool use (*39*) and enabled quick adaptation to new objects via meta-learning (*40*).

Suh and Tedrake (*41*) showed that a switched-linear pixel-based model can also yield strong performance. Transformer-based models have also been applied to robotic manipulation (*42*) and large-scale autonomous driving data (*43*). Finally, recent video diffusion models learn visual dynamics with improved scalability (*44–46*). These non-autoregressive models can not only predict future frames but also inpaint intermediate ones. Du *et al.* (*46*) use this property combined with an inverse dynamics model to perform control.

Pixel prediction methods can be readily applied to other data modalities as long as they can be represented as three-dimensional arrays, including depth (*47*), percepts from an optical

tactile sensor (*21*), and density fields of granular materials (*48*).

Pixel-space models are typically trained with maximum likelihood objectives in a self-supervised manner on video sequences, with prediction targets sampled from future frames. Agent action labels for action-conditioned models may be represented as end-effector poses or joint positions. These can be obtained through proprioception. When action labels are absent, some models infer latent action representations (*49–51*).

Pixel dynamics models are usually evaluated with metrics from the video generation literature. However, these metrics focus on visual appearance and often do not correlate with planning performance (*52*). Physical prediction-based benchmarks (*53*) partially bridge this gap, but developing additional metrics is an open challenge.

The wide availability of pixels may allow pixel-space models to achieve broad generalization capabilities. Pixel-prediction models have been trained on increasingly diverse datasets, for instance on robotic interactions across several robots and scenes (*54*). GAIA-1 (*43*) was trained using in-the-wild driving data, and UniSim (*45*) is a single model trained with robotic data, human videos, internet media, and navigation data.

Overall, pixel-based representations do not require explicit state estimation and, in principle, can model arbitrary physical phenomena. They bypass explicit perception pipelines but require large datasets to learn effectively in high-dimensional observation spaces. Convolutional neural networks (CNNs) are commonly used, with recent approaches employing Transformers and diffusion models. Despite these advances, such models often struggle with object permanence and temporal consistency, even when trained with substantial computational resources (*45, 46*). For control, pixel-based models are sensitive to partial observability, which can lead to hallucinations, and their high computational cost poses challenges for high-frequency control.

Latent Representations

Predicting dynamics from raw observations o_t is challenging due to their high dimensionality and redundancy. A common alternative is to project o_t into a lower-dimensional latent vector z_t first. Although both pixel-based models and latent-state models can be trained on pixel data, they differ in their prediction domain: pixel-based models autoregressively generate future observations, whereas latent-state models predict in abstract latent space.

This projection introduces inductive biases by assuming the state space admits a compact and smooth parameterization. This can substantially enhance learning efficiency and generalization by filtering out irrelevant variations.

Perception

A key challenge in learning a latent representation is ensuring the latent vector z_t encodes task-relevant features rather than collapsing to trivial solutions, such as mapping the set of all inputs to a constant vector. Existing approaches address this by imposing structure via supervision, and can be categorized into reconstruction-based and reconstruction-free methods.

Reconstruction-based training is a common approach for learning latent state representations that ensures that encoded states retain sufficient information to reconstruct raw observations. Early work, such as Embed to Control (*55*), enforced alignment between decoded and ground-truth observations using Kullback–Leibler divergence, but with the limiting assumption of linear state dependencies. More expressive models instead learn non-linear mappings with deep networks, such as variational autoencoders (*56*) and GNNs (*57*). Latent states trained to reconstruct volumetric scenes further impose strong geometric and 3D priors (*58, 59*). However, they are computationally expensive and impractical for real-time control.

In partially observable environments, reconstruction-based training extends to inferring occluded states. ACID (*60*) encodes partial RGB-D inputs into a 3D feature field, predicting occu-

pancy probabilities to handle occlusions for deformable object manipulation. When single-step observations are insufficient, recurrent models can aggregate history information (*7, 61*).

Reconstruction-based training may lead to latents encoding task-irrelevant details. To avoid this, reconstruction-free approaches use alternative learning signals. One alternative is to predict task-relevant features, such as object motion represented by optical flow (*14*). Inverse dynamics learning trains models to predict the action responsible for a state transition, ensuring only action-relevant latent features (*15*). Contrastive learning avoids trivial solutions, pulling predicted next states closer to ground truth states and pushing them away from incorrect encodings (*19*). When rewards are available, predicting rewards from latent representations provides compact, efficient, and task-relevant encodings (*62, 63*), at the cost of increased task dependence of the learned models.

Dynamics Learning

Latent dynamics models can be categorized as probabilistic or deterministic. Probabilistic models predict distributions over future states, whereas deterministic models estimate a single most likely outcome. In both cases, robot actions are typically incorporated by concatenating them with the estimated latent states inputting to the dynamics predictor.

Probabilistic models leverage classical statistical methods or deep neural networks to predict distributions over future states. Linear probabilistic models, such as Gaussian state-space models, are typically trained by optimizing distributional divergence metrics (*55, 64*). More expressive approaches are based on deep neural networks and often incorporate history information. DayDreamer (*7*) and DeformNet (*59*) apply recurrent state-space models (RSSMs) (*65*), which use recurrence to maintain temporal memory and capture uncertainty, to real-world manipulation. Probabilistic models can output multi-modal future predictions by combining mixture density networks with recurrent models (*8*).

When environmental dynamics are relatively predictable, deterministic models provide a simpler alternative. They may use multi-layer perceptrons (MLPs) for low-dimensional latent spaces (*15, 57, 58, 62, 63*) and CNNs for high-resolution feature maps (*14, 60*).

In summary, latent-state models have been applied to diverse manipulation settings, including rigid bodies, articulated objects, and fluids. Training objectives range from task-agnostic formulations, such as reconstruction, to highly task-specific losses incorporating reward signals. RSSMs and MLPs are commonly used for modeling dynamics in low-dimensional latent spaces, where well-structured representations often lead to sample-efficient learning. Task-specific objectives may produce representations that struggle to generalize, whereas task-agnostic approaches can support cross-task transfer, though generalization to varying object counts or scene configurations remains limited. Compact latent representations make these models computationally efficient, enabling fast closed-loop control.

Particle-Based Representations

Unlike latent and pixel-based representations, particle-based models explicitly encode 3D structure by representing objects as discrete points, capturing both surfaces and volumes. This structure enables precise interaction modeling, and incorporating strong physical priors improves sample efficiency.

Particles have long been used in physics-based simulation methods, such as MPM (*4*). These techniques underpin modern physics-based simulators (*31, 66*), but rely on approximate modeling, leading to a sim-to-real gap that often requires system identification (*32*). Learned particle dynamics models can predict particle behavior directly from real-world data.

Perception

In real-world applications, particles are commonly sampled from observed point clouds (*16, 17, 22, 67*). Single-camera methods can reconstruct point clouds through gradient-based optimization (*68*), though these are often noisy. Data is usually downsampled before training (*16, 17*). However, occlusion remains a challenge in cluttered environments. Some methods incorporate geometric priors to handle occlusion, for example assuming dough conforms to the shape of a tool interacting with it (*16*). Integrating tactile sensing may also improve particle state estimation in combination with historical observations and recurrent structures (*20*).

Alternatively, volumetric representations can be constructed from multi-view images via NeRF (*69*). Then, particles can be sampled from voxel grids using trilinear interpolation (*4*).

Dynamics Learning

Particle dynamics arise from local particle interactions, which models typically capture using inductive biases like spatial equivariance and locality. To this end, existing approaches primarily use graph-based architectures or convolutional models.

GNNs are widely used for modeling particle interactions. Particles are represented as graph nodes, and node features may include physical parameters or motion and displacement information. HRN (*70*) introduces a hierarchical graph structure where leaf particles encode local interactions, whereas root nodes provide object-level abstractions to handle rigid and non-rigid transformations. To enhance adaptability, DPI-Nets (*57, 71*) update dynamic interaction graphs during simulation, effectively capturing object deformations. This flexibility enabled DPI-Nets to lay the foundation for modeling elasto-plastic objects (*17, 72*), granular material manipulation (*22*), food preparation (*16*), and object packing (*20, 73*). GNS (*74, 75*) generalizes this framework by providing a simpler yet more accurate model for fluids, rigid bodies, and deformable materials.

Alternatively, convolutional architectures model local interactions without explicit graphs. SPNets (*76*) use specialized convolutions: ConvSP for particle-particle interactions and ConvSDF for differentiable collisions with static geometry. Actions are supplied by updating the poses of controllable objects, which then interact with other objects. Compared to graph-based models, convolutional architectures are often more efficient and parallelizable, but less flexible for long-range or irregular interactions.

To summarize, particle-based representations explicitly encode geometric structure, with physical properties preserved through particle interactions. They are particularly well-suited to deformable objects but have also been applied to rigid bodies and fluids. Estimating particle states from depth observations is sensitive to occlusions, and point tracking is often used to establish correspondences across frames. Graph-based networks and convolutional architectures are common modeling choices, offering strong inductive biases and sample efficiency. These inductive biases support generalization to novel object geometries, reflecting a trade-off: more demanding perception enables more accurate and efficient dynamics modeling through structured representations. For control, GNNs may face scalability challenges with dense graphs, whereas convolutional networks are generally lightweight. Particle representations also integrate multimodal inputs, such as vision and touch, to enable fine-grained control.

Keypoint-Based Representations

Keypoint representations consist of sparse points that may encode implicit or explicit semantic information. Unlike particle representations, which use dense 3D points to capture object geometries, keypoints offer a more compact state representation that retains only task-relevant points. Typically, keypoints are defined by a set of 2D or 3D coordinates; for instance, a rigid box can be represented by its eight corner points.

Unlike unordered particle sets, keypoints are often structured as ordered lists with semantic

meaning implicitly assigned to specific indices (e.g., visual features).

Perception

The literature presents three common approaches for keypoint extraction: supervised learning with manual labels, unsupervised learning using reconstruction losses, and zero-shot prediction using pre-trained vision models.

Supervised learning methods train networks to predict keypoints from labeled datasets, but efficient keypoint annotation remains a challenge. kPAM (*77*) addresses this by labeling keypoints in 3D and projecting them into image space. Dense Object Nets (*78*) introduce dense visual descriptors, tracking keypoints over time via feature similarity. This approach has been extended for keypoint-based object tracking (*13*) and deformable object tracking (*79*).

Unsupervised learning methods extract keypoints using encoder-decoder frameworks where the decoder reconstructs observations from keypoints. Transporter (*80*) is a representative 2D method, extracting keypoints from RGB images through feature inpainting and reconstruction. In 3D, KeypointDeformer (*81*) predicts shape-representative keypoints from object meshes, training on source-target mesh pairs to learn deformation consistency.

Recent work explores zero-shot keypoint detection using visual foundation models. RoboABC (*82*) aligns robot observations with human-object interaction data using CLIP and diffusion features (*83, 84*) to identify contact points. B2-3D (*85*) projects 2D DINO features into 3D space to detect category-specific keypoints with minimal annotations. KITE (*86*) extends this by training a grounding module to localize semantic keypoints based on text inputs.

Dynamics Learning

Keypoints can be processed similarly to particles using GNNs, since they also represent points in space (*87–90*). However, when organized as ordered lists, keypoints can encode additional semantic information. Thus, although GNNs are permutation-invariant and well-suited for un-

ordered data, ordered keypoints are typically processed with MLPs, which can leverage ordering information (*13, 91*). In both cases, actions are represented as fixed-dimensional vectors and concatenated with graph node features or keypoint feature vectors for dynamics prediction (*13, 88, 89*).

Keypoint-based representations focus on task-relevant features rather than full scene geometry, making them suitable for tasks where specific object regions are salient for control. They have been applied to both rigid-body and deformable object manipulation. Keypoints are typically extracted using learned detectors. Although more compact than particle sets, they are sensitive to occlusion and require consistent detection over time. Lightweight architectures such as MLPs or graph-based models are commonly used to capture keypoint dynamics and interactions. Because keypoints correspond to consistent abstract task-relevant structures, models can often generalize across object instances. Their compactness also enables fast inference, real-time planning, and feedback control.

Object-Centric Representations

A core challenge of scaling dynamics models to diverse scenes is the combinatorial complexity of possible object configurations in the world, which is challenging to handle without compositional generalization abilities. Humans address this by perceiving scenes in an object-centric way: containing discrete entities with boundaries and predictable interactions (*92*).

Motivated by this, some approaches adopt object-centric representations that model dynamics at the level of interacting objects rather than low-level particles or features. These structured representations support generalization to novel object arrangements, and are the highest abstraction level we consider for modeling dynamics.

Perception

Techniques for obtaining object-level latent representations from raw observations include segmenting objects from visual inputs and encoding their features, directly mapping multi-object scenes to structured object-centric encodings, or leveraging inverse rendering techniques to infer physical object states.

The first approach explicitly segments objects before extracting features. O2P2 (*93*) assumes access to instance segmentation and encodes each object separately, enforcing meaningful representations through a neural rendering engine. NS-DR (*94*) extends this to video, whereas RPIN (*95*) jointly detects and encodes objects for dynamics learning. Compositional NeRF (*23*) integrates segmentations across camera viewpoints for 3D consistency.

Alternatively, object-centric representations can be learned from multi-object scenes without explicit segmentation. Visual Interaction Networks (VIN) (*96*) extract object-wise latent representations from image sequences and decodes them into object states (e.g., position and velocity), but requires ground-truth supervision. To alleviate this, OP3 (*97*) performs unsupervised object discovery, iteratively refining posterior estimates of object assignments based on interaction data.

A third approach leverages inverse rendering to infer object states from raw observations. Tian *et al.* (*98*) use neural implicit object representations and optimization-based inference to estimate 6D object poses, achieving robust performance under varying lighting conditions.

Dynamics Learning

Object-centric dynamics models treat objects as discrete entities and model their interactions. They can be implemented using generic neural networks or graph-based architectures that explicitly leverage relational structures.

To explicitly leverage relational structure, Neural Physics Engine (NPE) (*99*) introduces a

mechanism akin to message passing in GNNs. It iterates over object pairs, predicting their relative motion, and aggregating predictions. O2P2 (*93*) and OP3 (*97*) adopt similar object-centric architectures. Although O2P2 incorporates environment actions such as object placement or motion, OP3 further embeds action information as a latent vector to modulate both per-object dynamics and interactions.

GNNs provide a more structured approach to modeling object interactions through iterative message passing. The foundational Interaction Networks (IN) (*100*) represent objects as graph nodes and encode relational attributes (e.g., restitution coefficients, spring constants) in the edges. However, the model assumes access to ground-truth physical properties such as shape and mass. When object states are unavailable, latent representations or robot actions can act as node features (*23, 94*).

In summary, object-centric representations are well-suited for tasks involving multi-object interactions but less effective for modeling continuous materials such as fluids or highly deformable objects. Perception is typically achieved through instance segmentation, inverse rendering, or object proposal techniques, though it remains challenging in general. Dynamics are often modeled using graph-based architectures, enabling relational reasoning and modularity. With object-level priors, these models support combinatorial generalization across varying numbers and configurations of objects. They are generally computationally efficient for control, but perception can become a bottleneck, potentially introducing latency.

Comparing and Selecting State Representations

Each state representation provides distinct trade-offs in modeling capacity, sample efficiency, generalization, task alignment, interpretability, and computational cost (Figure 3).

Less structured representations, such as latent states and pixels, simplify state estimation but introduce challenges in model bias and generalization. They often require larger datasets and

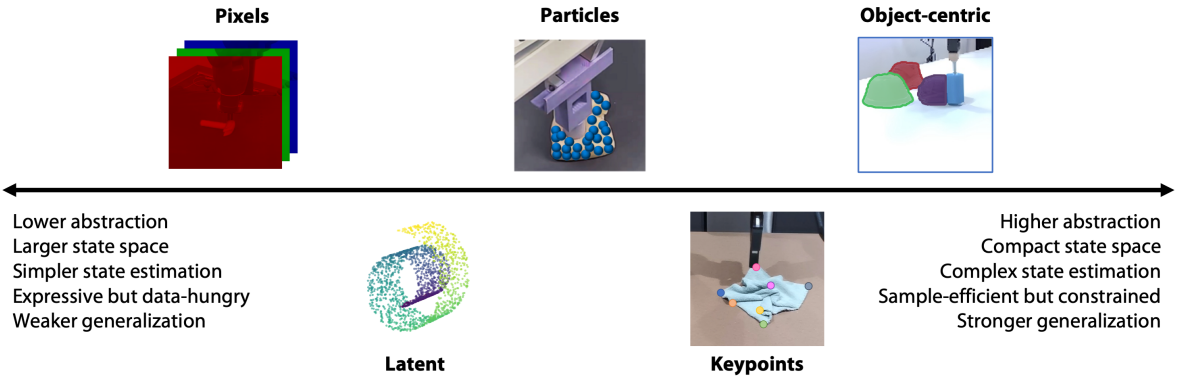

Figure 3: **A spectrum of state representations with varying structure priors.** State representations in dynamics models range from unstructured (pixels, latent) to structured (particles, keypoints, object-centric). Increasing structure introduces stronger priors and abstraction, enabling better generalization but requiring more complex state estimation. The "Swiss Roll" illustration for latent states is inspired by Tenenbaum *et al.* (*101*).

can hallucinate predictions in out-of-distribution scenarios. These inaccuracies can be problematic during downstream control, as optimization over unreliable model outputs may degrade control performance.

Conversely, structured representations such as particles and object-centric models provide strong inductive biases, like spatial equivariance, enhancing generalization and prediction accuracy. These models are often trainable within a few GPU hours (*17, 71*) and more robust for policy and trajectory optimization. However, they pose challenges for state estimation. Particle-based methods require temporally consistent 3D point tracks, which remain difficult to obtain (*90*). Similarly, object-centric representations depend on accurate perception modules, often making state estimation a bottleneck for scaling structured models.

The optimal choice of state representation depends on both the downstream task and hardware constraints. For instance, object-centric representations excel in manipulation tasks involving multiple rigid bodies (*94, 96, 102*) such as object rearrangement (*23*), stacking (*93*), and sliding (*14*) due to their high level of abstraction, but are unsuited for fluids or granular materials

for which it is unclear how to define the notion of an object. Particle-based representations flexibly capture the dynamics of deformable and non-rigid objects (*71, 74*), including dough (*17*), cloth (*103*), and soft toys (*20*), though they often require multi-view RGB-D sensing for point cloud perception. Latent and pixel-based models (*44–46*) can in principle handle arbitrary entities. However, they may produce physically inconsistent predictions without physics priors and explicit 3D representations, particularly in contact-rich scenarios such as cutting or splitting objects. Additionally, highly specular, transparent, or otherwise visually complex materials can be challenging for RGB image reconstruction.

Interpretability is another key consideration. For some state representations, such as pixels, particles, and keypoints, visualizing predicted trajectories is natural, making it simple to diagnose failure cases and refine models. In contrast, reconstruction-free latent-state models often lack this transparency.

Finally, computational cost is crucial for real-world robotic deployment. Pixel-based models require high-capacity architectures, whereas keypoint-based models often operate with smaller networks. Graph-based methods, often used with particle representations, can scale linearly with the number of graph edges or quadratically with the particle count, making inference costly. Additionally, backpropagation through models during gradient-based action optimization can add significant computational overhead.

Table 1 summarizes the key trade-offs across state representations, including sensing modalities, computational requirements, and target applications. Although existing methods excel at handling specific object types and sensor inputs, a unifying representation that generalizes across diverse robotic tasks remains a challenging but crucial direction for future work.

State Repre.	References	Sensing				Dynamics		Object Type			# Obj.
		RGB	Depth	Multi-View	Tactile	2D	3D	Rigid	Deformable	Fluids	Multiple
Latent State	Agrawal *et al.* (*15*)	✓					✓	✓			
	Yan *et al.* (*19*)	✓				✓			✓		
	Wu *et al.* (*7*)	✓					✓	✓			
	Li *et al.* (*58*)	✓		✓			✓	✓	✓	✓	✓
	Shen *et al.* (*60*)	✓	✓				✓	✓	✓		
Pixel	Finn *et al.* (*11*)	✓				✓		✓	✓		✓
	Suh *et al.* (*41*)	✓				✓		✓			
	Hoque *et al.* (*47*)	✓	✓				✓		✓		
	Du *et al.* (*46*)	✓				✓		✓			✓
	Yang *et al.* (*45*)	✓				✓		✓			✓
Particles	Gonzalez *et al.* (*74*)						✓		✓		✓
	Li *et al.* (*71*)		✓				✓		✓	✓	✓
	Ai *et al.* (*20*)	✓	✓	✓	✓		✓	✓	✓		✓
	Shi *et al.* (*16*)	✓	✓	✓			✓		✓		
	Wang *et al.* (*22*)	✓	✓			✓		✓			✓
Keypoints	Manuelli *et al.* (*13*)	✓	✓				✓	✓			
	Wang *et al.* (*91*)	✓					✓	✓			✓
	Li *et al.* (*87*)	✓				✓			✓		✓
	Ma *et al.* (*88*)		✓			✓			✓		
	Rezazadeh *et al.* (*89*)	✓				✓		✓			✓
Object-centric	Watters *et al.* (*96*)	✓				✓		✓			✓
	Janner *et al.* (*93*)	✓					✓	✓			✓
	Xu *et al.* (*14*)	✓	✓				✓	✓			✓
	Yi *et al.* (*94*)	✓					✓	✓			✓
	Driess *et al.* (*23*)	✓	✓				✓	✓	✓		✓

Table 1: **Summary of key studies on dynamics learning.** This table categorizes the literature based on the type of state representation, sensors used, dynamics modeled, and object types considered. The dimensions of the dynamics (2D, 3D) refer to the space in which object rotations, translations, and deformations are modeled.

Connection to Robotic Control

Learning-based dynamics models can be integrated with control modules to generate robot motions for predefined task objectives. We first detail two ways to leverage learned dynamics models, and then discuss representative tasks that benefit from this integration.

Control Methods

Control strategies using learning-based dynamics models fall into two main paradigms: motion planning and policy learning.

Motion Planning

Motion planning searches for a feasible path from an initial state to a goal state while satisfying task constraints. Learned dynamics models enable planning in complex or unknown environments, where analytical models are unavailable, inaccurate, or hard to obtain.

Path Planning. Path planning focuses on finding a sequence of collision-free states, without modeling system dynamics. Sampling-based methods such as RRT (*104*) and PRM (*105*) are widely used to search high-dimensional spaces with complex constraints. RRT incrementally expands a search tree through random sampling. The resulting paths can then be refined into dynamically feasible trajectories through trajectory optimization with learned dynamics models.

Trajectory Optimization. Trajectory optimization refines action sequences locally to improve task performance, directly leveraging learned dynamics models to simulate and evaluate outcomes. Sampling-based methods like CEM (*106*) and MPPI (*107*) explore multiple action candidates, while gradient-based methods adjust actions using cost gradients enabled by model differentiability. Moreover, learned dynamics models can be integrated with online system identification to adapt to uncertain dynamics (*14, 20, 108*).

Policy Learning

In contrast to motion planning, policy learning seeks to directly obtain a map from observations to actions. Learned dynamics models can provide simulated transitions as training data.

Supervised Learning. One approach is to generate training data in the form $\langle s_t, s_g, a_t \rangle$, where the state s_t and action a_t yield the next state s_g. An inverse dynamics model can then learn to predict the action needed to transition from s_t to s_g, acting as a goal-conditioned policy. However, errors may accumulate over extended rollouts, and multi-modal action distributions can be hard to fit, since multiple actions could achieve the same transition.

Reinforcement Learning. Reinforcement learning (RL) optimizes policies through trial-and-error interactions to maximize cumulative rewards. Learned dynamics models facilitate this process by simulating transitions (*109*), allowing policies to be trained with a reduced or negligible number of real-environment interactions. However, inaccuracies in the learned dynamics model can lead to policy exploitation, particularly in state distributions not well-supported by training data. This can be mitigated by fine-tuning the policy on real-world data in addition to simulated rollouts (*110*).

Representative Robotic Tasks

Learning-based dynamics models have been applied across tasks from object pushing to deformable and multi-object manipulation. This section highlights key applications and integrations with motion planning and policy learning techniques.

Object Repositioning. Object repositioning is widely used to evaluate learned dynamics models in robotic control. Latent representations (*15, 60*) and pixels (*21*) have been used to

represent single-object scenes, while keypoints can serve as a lower-dimensional representation for efficient dynamics learning given the limited degrees of freedom of rigid objects (*13*). On the other hand, multi-object scenarios can be better modeled with object-centric representations (*23*). For control, motion planning methods such as random search (*14, 38, 60*), MPPI (*13, 107*), and CEM (*11, 106*) can optimize action sequences. Online system identification can help handle objects with unknown physical properties (*20*). Alternatively, inverse dynamics models trained alongside forward models can directly infer actions from current and target states (*15*).

Deformable Object Manipulation. Deformable object manipulation presents challenges due to high-dimensional shape variations and complex contact dynamics (*24, 25*). Particle-based representations can capture the arbitrary geometries of deformable objects (*16,17,20,71,74,108*) and can be abstracted into keypoints for objects with salient features such as cloth (*88*). Learned dynamics models have been integrated with trajectory optimization for manipulating rope (*19, 23*), cloth (*59*), dough (*17, 111*), and soft toys (*20*). These models also enable training goal-conditioned policies, as demonstrated in long-horizon tasks such as making dumplings (*16*).

Multi-Object Manipulation. Manipulation involving multiple objects requires efficient planning to manage large state spaces. Particle-based (*20*) and object-centric representations (*23*) perform well in multi-object modeling, whereas pixel-based methods struggle with modeling contact-rich interactions. To perform control using learned dynamics models, RoboPack (*20*) applies MPPI with action priors for object insertion. Latent-space RRT has been combined with model predictive control for long-term planning and real-time corrections (*23*).

Tool-Use Manipulation. Modeling tool-use dynamics may extend robotic capabilities beyond manipulating objects directly with an end-effector. Particles can provide a unified repre-

sentation for objects, tools, and robot end effectors (*16, 20, 22*), but require detailed 3D sensing; pixel-based methods offer a lightweight perception alternative (*39*). Learned models have been used for shaping dough with rollers and punches (*16*), non-prehensile box manipulation with compliant tools (*20*), and granular material manipulation (*22*). For extended tasks requiring tool selection and task execution, action proposal models improve planning efficiency for sampling-based planning (*39*).

Figure 4 illustrates these tasks, and a summary of the discussed work is provided in Table 2.

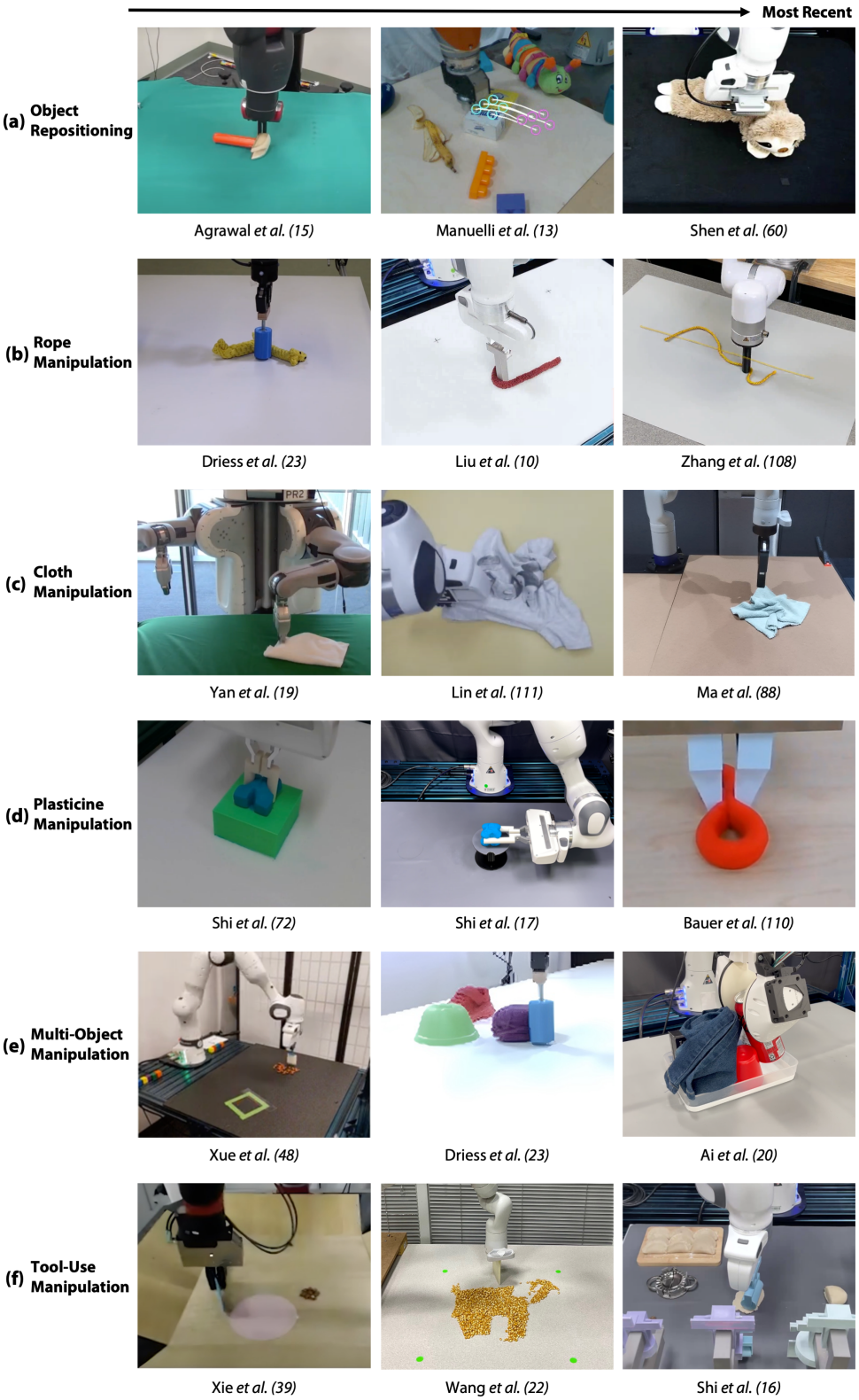

Figure 4: **Robotic manipulation tasks enabled with learning-based dynamics models.** (a) object repositioning, (b) rope manipulation, (c) cloth manipulation, (d) plasticine manipulation, (e) multi-object manipulation, and (f) tool-use manipulation. Examples span rigid and deformable objects, multi-object settings, and tool-assisted operations.

Task	References	Representation	Dynamics Model Class	Control
Object Repositioning	Agarwal *et al.* (*15*)	Latent	CNN	Greedy planner
	Shen *et al.* (*60*)	Latent	MLP	Random search
	Tian *et al.* (*21*)	Pixel	CNN	CEM
	Manuelli *et al.* (*13*)	Keypoint	GNN	MPPI
	Driess *et al.* (*23*)	Object-centric	GNN	RRT, random search
Rope Manipulation	Yan *et al.* (*19*)	Latent	CNN	Random search
	Zhang *et al.* (*108*)	Particle	GNN	MPPI
	Ma *et al.* (*88*)	Keypoint	GNN	Random search
	Liu *et al.* (*10*)	Keypoint	MLP	Mixed-Integer Programming
	Driess *et al.* (*23*)	Object-centric	GNN	RRT, random search
Cloth Manipulation	Yan *et al.* (*19*)	Latent	CNN	Random search
	Hoque *et al.* (*47*)	Pixel	CNN	CMA-ES
	Lin *et al.* (*112*)	Particle	GNN	Random search
	Ma *et al.* (*88*)	Keypoint	GNN	Random search
Plasticine Manipulation	Shi *et al.* (*17*)	Particle	GNN	Gradient descent
	Shi *et al.* (*16*)	Particle	GNN	Learned policy
	Bauer *et al.* (*111*)	Latent	Transformer	CEM
Multi-Object Manipulation	Xie *et al.* (*39*)	Pixel	CNN, LSTM	CEM
	Xue *et al.* (*48*)	Pixel	CNN	Gradient descent
	Ai *et al.* (*20*)	Particle	GNN	MPPI
	Rezazadeh *et al.* (*89*)	Keypoint	MLP	GraphMPC
	Driess *et al.* (*23*)	Object-centric	GNN	RRT, random search
Tool-Use Manipulation	Xie *et al.* (*39*)	Pixel	CNN, LSTM	CEM
	Shi *et al.* (*16*)	Particle	GNN	Learned policy
	Wang *et al.* (*22*)	Particle	GNN	Gradient descent
	Ai *et al.* (*20*)	Particle	GNN	MPPI

Table 2: **Summary of robotic tasks achieved by integrating learning-based dynamics models with planning.** The table presents the designs, including representation, dynamics model class, and control methods used for various robotic tasks. Each row represents a specific task and highlights the combination of approaches used to tackle it.

Future Directions

Learning-based dynamics models have advanced adaptive control in robotics model-based planning and policy learning. However, current systems remain far from human-level generalization, adaptability, and robustness in unstructured environments. This section discusses key limitations and outlines promising directions for future research.

State Estimation

Partially Observable Domains. Real-world environments are inherently partially observable due to visual occlusions and unknown physical properties like material rigidity and friction. Although passive history (*7, 57*), active perception (*14*), and multi-modal sensing (*20*) improve state estimation, challenges remain in cluttered and unstructured scenes. Structured representations such as particles require precise perception capabilities, whereas less structured models like pixels avoid this but often struggle with accuracy and generalization, especially for contact dynamics (*45*). Future work should explore new representations and robust state estimation methods to better handle partial observability.

Multi-Modal Perception. Although most prior work relies on visual sensing, other modalities, including tactile (*20*) and audio sensing (*113*), provide complementary information for perception and control. However, integrating multi-modal signals introduces several challenges. Differences in statistical distributions across modalities complicate model training, while mismatched sensing frequencies create deployment-time difficulties. Additionally, effectively fusing heterogeneous signals into a unified representation remains an open problem. Addressing these challenges will enable more robust dynamics reasoning and control performance across a wider range of tasks.

Dynamics Learning

Robust Dynamics Models. Inaccuracies in learned dynamics models can be exploited by planning and reinforcement learning agents, leading to failures in long-horizon tasks. Ensuring robust predictions across the entire state-action space is challenging due its combinatorial size. Additionally, certain state-action subspaces can be difficult or unsafe to explore, limiting counterfactual reasoning capabilities. Strategies to address this include using simulation data to cover challenging regions of the state-action space, introducing physics priors to reduce data requirements (*114*), and using probabilistic models to account for aleatoric uncertainty (*7, 59*).

Foundation Dynamics Models. Recent advances in foundation models (*115*) highlight the potential of large-scale training for broadly capable vision and language models. In contrast, most learned dynamics models remain narrow-domain due to the lack of large-scale real-world datasets with action labels (*19, 103*). Scaling up dynamics models may require inferring actions from unlabeled data like internet videos. Early efforts, such as learning latent actions (*51, 116*), suggest promising directions for this goal.

Dynamics Priors from Foundation Models. Estimating physical properties such as mass, friction, and deformability is crucial for accurate dynamics modeling, yet remains challenging. Prior work has attempted to infer these properties from observations, using visual cues (*98*), tactile sensing (*20*), or multi-view depth images (*14*). Recent foundation models demonstrate commonsense reasoning about material properties (e.g., a sofa deforms under pressure) (*117, 118*), offering a potential source of priors for estimating system parameters (*108*). By integrating these priors with learned dynamics models, future work could reduce reliance on real-world data and online identification.

Emerging Representations from Graphics Research. Scene representations from computer graphics offer new possibilities for dynamics learning. NeRF-based representations address limitations of direct pixel representations by capturing multi-view consistency and 3D structure (*23, 58*). Particle-based models tend to struggle with smooth continuous deformations, but recent advances in 3D Gaussian Splatting (3DGS) (*119*) may help address this by modeling particles as Gaussian functions, producing smoother and more flexible surfaces. Although 3DGS has been applied to dynamic scene reconstruction (*120, 121*), its integration with action-conditioned dynamics models remains underexplored. Early efforts include tracking objects with 3DGS in particle-based models (*122*), but deeper integration is a promising future direction.

Large-Scale Scene Representations. Most learning-based dynamics models focus on small-scale tabletop environments and local interactions (*10, 88, 111*), limiting their applicability to real-world tasks that require reasoning over large, dynamic spaces. Traditional approaches, such as Simultaneous Localization and Mapping (SLAM) (*123*), provide global geometric maps but lack dynamic information. Future directions include developing scene representations at varying levels of abstractions that capture both global structure and local interactions, training dynamics models from local interactions while maintaining scene-level coherence, and designing efficient update mechanisms that modify only affected scene regions (*124*).

Robotic Control

Hierarchical Dynamics Modeling and Planning. Highly detailed dynamics models are not always ideal for planning, as they induce large search spaces and high computational cost. Instead, modeling environments at multiple levels of abstraction can enable more efficient hierarchical planning (*125*) for long-horizon tasks. Low-level models can capture fine-grained

physical interactions for motor control, whereas high-level models may represent skill-level transitions or abstract state dynamics to support task planning (*126*). The most effective models are often those that are sufficient for the decision at hand (i.e., effective) while remaining minimal in complexity (i.e., efficient). Future work may explore constructing such dynamics models across different abstraction levels: spatially, from particle-based to object-centric representations, and temporally, from short-horizon physical transitions to extended skill executions. An exciting direction is to investigate how to learn unified hierarchical dynamics models or to compose and interface separate models at different abstraction levels, and integrate them with hierarchical planning frameworks to support decision-making across large spatial and temporal scales.

Learning to Plan. Existing work typically obtains locally accurate dynamics models and restricts exploration to well-supported regions through engineered action spaces or carefully designed planning costs. Machine learning offers a way to automate this process. One line of work focuses on improving planning efficiency by learning heuristics to guide search (*127*) or optimizing surrogate objectives such as action space selection (*128*). Another approach is to alleviate the need for globally accurate dynamics models by learning action generative models that constrain the sampling space during planning (*129*). Despite these initial explorations, learning to identify reliable regions of learned dynamics models and to plan efficiently and robustly in the presence of model imperfections remains an open challenge.

Performance Guarantees. Prediction errors in learned dynamics models can accumulate over time and degrade planning performance. Uncertainty quantification methods, such as Bayesian neural networks (*130*) and variational inference (*131*), help mitigate this by additionally providing confidence estimates. Although uncertainty estimates have been used in model-based RL (*110*) and trajectory optimization (*132*), their use in robust planning remains under-

explored. Coupling these techniques with theoretical guarantees from planners is a promising path towards reliable real-world deployment.

Conclusions

Learning-based dynamics models have substantially advanced robotic capabilities, from simple tasks to more complex scenarios involving long-horizon planning and deformable objects. The choice of state representation critically influences a dynamics model's accuracy, data efficiency, and state estimation requirements. This review has presented a robotics-centric examination of dynamics models, emphasizing their integration with perception and control. Despite recent advances, key challenges remain in developing robust, generalizable, and scalable dynamics models, which could serve as foundational tools for robotic manipulation. Fundamental questions persist: What representations best capture diverse scenes? How can inductive biases balance expressiveness and generalization? Addressing these questions is essential for advancing adaptive, interpretable, and robust robotic systems.

References

1. J. B. Tenenbaum, C. Kemp, T. L. Griffiths, N. D. Goodman, *How to Grow a Mind: Statistics, Structure, and Abstraction*, *Science* **331**, 1279 (2011).

2. P. W. Battaglia, J. B. Hamrick, J. B. Tenenbaum, *Simulation as an engine of physical scene understanding*, *Proceedings of the National Academy of Sciences* **110**, 18327 (2013).

3. M. Müller, B. Heidelberger, M. Hennix, J. Ratcliff, *Position based dynamics*, *Journal of Visual Communication and Image Representation* **18**, 109 (2007).

4. D. Sulsky, S.-J. Zhou, H. L. Schreyer, *Application of a particle-in-cell method to solid mechanics*, *Computer Physics Communications* **87**, 236 (1995).

5. A. Ajay, *et al.*, *Combining Physical Simulators and Object-Based Networks for Control*, *IEEE International Conference on Robotics and Automation (ICRA)* (2019).

6. A. Ajay, *et al.*, *Augmenting Physical Simulators with Stochastic Neural Networks: Case Study of Planar Pushing and Bouncing*, *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2018).

7. P. Wu, A. Escontrela, D. Hafner, P. Abbeel, K. Goldberg, *DayDreamer: World Models for Physical Robot Learning*, *Conference on Robot Learning (CoRL)* (2022).

8. D. Ha, J. Schmidhuber, *Recurrent World Models Facilitate Policy Evolution*, *Advances in Neural Information Processing Systems (NeurIPS)* (Curran Associates, Inc., 2018).

9. Y. Li, H. He, J. Wu, D. Katabi, A. Torralba, *Learning Compositional Koopman Operators for Model-Based Control*, *International Conference on Learning Representations (ICLR)* (2020).

10. Z. Liu, *et al.*, *Model-Based Control with Sparse Neural Dynamics*, *Advances in Neural Information Processing Systems (NeurIPS)* (2023).

11. C. Finn, S. Levine, *Deep visual foresight for planning robot motion*, *IEEE International Conference on Robotics and Automation (ICRA)* (2017).

12. L. P. Kaelbling, *The foundation of efficient robot learning*, *Science* **369**, 915 (2020).

13. L. Manuelli, Y. Li, P. R. Florence, R. Tedrake, *Keypoints into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning*, *Conference on Robot Learning (CoRL)* (2020).

14. Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, S. Song, *DensePhysNet: Learning Dense Physical Object Representations Via Multi-Step Dynamic Interactions*, *Robotics: Science and Systems* (2019).

15. P. Agrawal, A. Nair, P. Abbeel, J. Malik, S. Levine, *Learning to poke by poking: experiential learning of intuitive physics*, *Advances in Neural Information Processing Systems (NeurIPS))* (2016).

16. H. Shi, H. Xu, S. Clarke, Y. Li, J. Wu, *RoboCook: Long-Horizon Elasto-Plastic Object Manipulation with Diverse Tools*, *Conference on Robot Learning (CoRL)* (2023).

17. H. Shi, H. Xu, Z. Huang, Y. Li, J. Wu, *RoboCraft: Learning to see, simulate, and shape elasto-plastic objects in 3D with graph networks*, *The International Journal of Robotics Research* **43**, 533 (2024).

18. A. Longhini, *et al.*, *EDO-Net: Learning Elastic Properties of Deformable Objects from Graph Dynamics*, *IEEE International Conference on Robotics and Automation (ICRA)* (2023).

19. W. Yan, A. Vangipuram, P. Abbeel, L. Pinto, *Learning Predictive Representations for Deformable Objects Using Contrastive Estimation*, *Conference on Robot Learning (CoRL)* (2020).

20. B. Ai, *et al.*, *RoboPack: Learning Tactile-Informed Dynamics Models for Dense Packing*, *Robotics: Science and Systems* (2024).

21. S. Tian, *et al.*, *Manipulation by Feel: Touch-Based Control with Deep Predictive Models*, *IEEE International Conference on Robotics and Automation (ICRA)* (2019).

22. Y. Wang, Y. Li, K. D. Campbell, L. Fei-Fei, J. Wu, *Dynamic-Resolution Model Learning for Object Pile Manipulation*, *Robotics: Science and Systems* (2023).

23. D. Driess, Z. Huang, Y. Li, R. Tedrake, M. Toussaint, *Learning Multi-Object Dynamics with Compositional Neural Radiance Fields*, *Conference on Robot Learning (CoRL)* (2022).

24. H. Yin, A. Varava, D. Kragic, *Modeling, learning, perception, and control methods for deformable object manipulation*, *Science Robotics* **6** (2021).

25. A. Longhini, *et al.*, *Unfolding the Literature: A Review of Robotic Cloth Manipulation*, *arXiv:2407.01361* (2024).

26. C. K. Liu, D. Negrut, *The Role of Physics-Based Simulators in Robotics*, *Annu. Rev. Control. Robotics Auton. Syst.* **4**, 35 (2021).

27. R. Newbury, *et al.*, *A Review of Differentiable Simulators*, *IEEE Access* (2024).

28. J. R. Kubricht, K. J. Holyoak, H. Lu, *Intuitive Physics: Current Research and Controversies*, *Trends in Cognitive Sciences* **21**, 749 (2017).

29. K. M. Lynch, H. Maekawa, K. Tanie, *Manipulation and Active Sensing by Pushing Using Tactile Feedback*, *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (1992).

30. E. Todorov, T. Erez, Y. Tassa, *MuJoCo: A physics engine for model-based control*, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2012).

31. V. Makoviychuk, *et al.*, *Isaac gym: High performance gpu-based physics simulation for robot learning*, *arXiv preprint arXiv:2108.10470* (2021).

32. W. Zhao, J. P. Queralta, T. Westerlund, *Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey*, *IEEE Symposium Series on Computational Intelligence* (2020).

33. M. Bauza, A. Rodriguez, *A probabilistic data-driven model for planar pushing*, *IEEE International Conference on Robotics and Automation (ICRA)* (2017).

34. S. Pfrommer, M. Halm, M. Posa, *ContactNets: Learning discontinuous contact dynamics with smooth, implicit representations*, *Conference on Robot Learning (CoRL)* (2020).

35. K. R. Allen, *et al.*, *Graph network simulators can learn discontinuous, rigid contact dynamics*, *Conference on Robot Learning (CoRL)* (2022).

36. K. Allen, *et al.*, *Inverse Design for Fluid-Structure Interactions using Graph Network Simulators*, *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

37. T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. W. Battaglia, *Learning Mesh-Based Simulation with Graph Networks*, *International Conference on Learning Representations (ICLR)* (2021).

38. F. Ebert, *et al.*, *Visual foresight: Model-based deep reinforcement learning for vision-based robotic control*, *arXiv:1812.00568* (2018).

39. A. Xie, F. Ebert, S. Levine, C. Finn, *Improvisation through physical understanding: Using novel objects as tools with visual foresight* (2019).

40. L. Yen-Chen, M. Bauza, P. Isola, *Experience-embedded Visual Foresight*, *Conference on Robot Learning (CoRL)* (2019).

41. H. T. Suh, R. Tedrake, *The surprising effectiveness of linear models for visual foresight in object pile manipulation*, *Algorithmic Foundations of Robotics* (2021).

42. A. Gupta, *et al.*, *MaskViT: Masked Visual Pre-Training for Video Prediction*, *International Conference on Learning Representations (ICLR)* (2023).

43. A. Hu, *et al.*, *GAIA-1: A generative world model for autonomous driving*, *arXiv:2309.17080* (2023).

44. Y. Du, *et al.*, *Video Language Planning*, *International Conference on Learning Representations (ICLR)* (2024).

45. M. Yang, *et al.*, *Learning interactive real-world simulators*, *International Conference on Learning Representations (ICLR)* (2024).

46. Y. Du, *et al.*, *Learning universal policies via text-guided video generation*, *Advances in Neural Information Processing Systems (NeurIPS)* (2023).

47. R. Hoque, *et al.*, *VisuoSpatial Foresight for physical sequential fabric manipulation*, *Auton. Robots* **46**, 175 (2022).

48. S. Xue, S. Cheng, P. Kachana, D. Xu, *Neural Field Dynamics Model for Granular Object Piles Manipulation*, *Conference on Robot Learning (CoRL)* (2023).

49. O. Rybkin, K. Pertsch, K. G. Derpanis, K. Daniilidis, A. Jaegle, *Learning what you can do before doing anything*, *International Conference on Learning Representations (ICLR)* (2019).

50. K. Schmeckpeper, *et al.*, *Learning Predictive Models from Observation and Interaction*, *European Conference on Computer Vision (ECCV)* (2020).

51. J. Bruce, *et al.*, *Genie: Generative Interactive Environments*, *International Conference on Machine Learning (ICML)* (2024).

52. S. Tian, C. Finn, J. Wu, *A Control-Centric Benchmark for Video Prediction*, *International Conference on Learning Representations (ICLR)* (2023).

53. D. M. Bear, *et al.*, *Physion: Evaluating physical prediction from vision in humans and machines*, *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

54. S. Dasari, *et al.*, *RoboNet: Large-Scale Multi-Robot Learning*, *Conference on Robot Learning (CoRL)* (2019).

55. M. Watter, J. T. Springenberg, J. Boedecker, M. A. Riedmiller, *Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images*, *Advances in Neural Information Processing Systems (NeurIPS)* (2015).

56. D. P. Kingma, M. Welling, *Auto-Encoding Variational Bayes*, *International Conference on Learning Representations (ICLR)* (2014).

57. Y. Li, *et al.*, *Propagation Networks for Model-Based Control Under Partial Observation*, *IEEE International Conference on Robotics and Automation (ICRA)* (2019).

58. Y. Li, S. Li, V. Sitzmann, P. Agrawal, A. Torralba, *3D Neural Scene Representations for Visuomotor Control*, *Conference on Robot Learning (CoRL)* (2021).

59. C. Li, *et al.*, *DeformNet: Latent Space Modeling and Dynamics Prediction for Deformable Object Manipulation*, *IEEE International Conference on Robotics and Automation (ICRA)* (2024).

60. B. Shen, *et al.*, *Action-conditional implicit visual dynamics for deformable object manipulation*, *The International Journal of Robotics Research (IJRR)* (2024).

61. M. J. Hausknecht, P. Stone, *Deep Recurrent Q-Learning for Partially Observable MDPs*, *AAAI Fall Symposium Series* (2015).

62. N. Hansen, H. Su, X. Wang, *Temporal Difference Learning for Model Predictive Control*, *International Conference on Machine Learning (ICML)* (2022).

63. N. Hansen, H. Su, X. Wang, *TD-MPC2: Scalable, Robust World Models for Continuous Control*, *International Conference on Learning Representations (ICLR)* (2024).

64. B. Lusch, J. Kutz, S. Brunton, *Deep learning for universal linear embeddings of nonlinear dynamics*, *Nature Communications* **9** (2018).

65. D. Hafner, *et al.*, *Learning Latent Dynamics for Planning from Pixels*, *International Conference on Machine Learning (ICML)* (2019).

66. F. Xiang, *et al.*, *SAPIEN: A SimulAted Part-based Interactive ENvironment*, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

67. X. Lin, *et al.*, *Planning with Spatial-Temporal Abstraction from Point Clouds for Deformable Object Manipulation*, *Conference on Robot Learning (CoRL)* (2022).

68. S. Chen, X. Ma, Y. Lu, D. Hsu, *Ab Initio Particle-based Object Manipulation*, *Robotics: Science and Systems* (2021).

69. B. Mildenhall, *et al.*, *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, *European Conference on Computer Vision (ECCV)* (2020).

70. D. Mrowca, *et al.*, *Flexible Neural Representation for Physics Prediction*, *Advances in Neural Information Processing Systems (NeurIPS)* (2018).

71. Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, A. Torralba, *Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids*, *International Conference on Learning Representations (ICLR)* (2019).

72. H. Shi, H. Xu, Z. Huang, Y. Li, J. Wu, *RoboCraft: Learning to See, Simulate, and Shape Elasto-Plastic Objects with Graph Networks*, *Robotics: Science and Systems* (2022).

73. H. Chen, *et al.*, *Predicting Object Interactions with Behavior Primitives: An Application in Stowing Tasks*, *Conference on Robot Learning (CoRL)* (2023).

74. A. Sanchez-Gonzalez, *et al.*, *Learning to Simulate Complex Physics with Graph Networks*, *International Conference on Machine Learning (ICML)* (2020).

75. N. Tuomainen, D. Blanco-Mulero, V. Kyrki, *Manipulation of Granular Materials by Learning Particle Interactions*, *IEEE Robotics and Automation Letters* **7**, 5663 (2022).

76. C. Schenck, D. Fox, *SPNets: Differentiable fluid dynamics for deep neural networks*, *Conference on Robot Learning (CoRL)* (2018).

77. L. Manuelli, W. Gao, P. Florence, R. Tedrake, *kPAM: Keypoint affordances for category-level robotic manipulation*, *The International Symposium of Robotics Research* (2019).

78. P. R. Florence, L. Manuelli, R. Tedrake, *Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation*, *Conference on Robot Learning (CoRL)* (2018).

79. J. Grannen, *et al.*, *Untangling Dense Knots by Learning Task-Relevant Keypoints*, *Conference on Robot Learning (CoRL)* (2021).

80. T. D. Kulkarni, *et al.*, *Unsupervised learning of object keypoints for perception and control*, *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

81. T. Jakab, *et al.*, *KeypointDeformer: Unsupervised 3D keypoint discovery for shape control*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

82. Y. Ju, *et al.*, *Robo-ABC: Affordance Generalization Beyond Categories via Semantic Correspondence for Robot Manipulation*, *European Conference on Computer Vision (ECCV)* (2024).

83. A. Radford, *et al.*, *Learning transferable visual models from natural language supervision*, *International Conference on Machine Learning (ICML)* (2021).

84. L. Tang, M. Jia, Q. Wang, C. P. Phoo, B. Hariharan, *Emergent Correspondence from Image Diffusion*, *Advances in Neural Information Processing Systems (NeurIPS)* (2023).

85. T. Wimmer, P. Wonka, M. Ovsjanikov, *Back to 3D: Few-Shot 3D Keypoint Detection with Back-Projected 2D Features*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

86. P. Sundaresan, S. Belkhale, D. Sadigh, J. Bohg, *KITE: Keypoint-Conditioned Policies for Semantic Manipulation*, *Conference on Robot Learning (CoRL)* (2023).

87. Y. Li, A. Torralba, A. Anandkumar, D. Fox, A. Garg, *Causal Discovery in Physical Systems from Videos*, *Advances in Neural Information Processing Systems* (2020).

88. X. Ma, D. Hsu, W. S. Lee, *Learning Latent Graph Dynamics for Visual Manipulation of Deformable Objects*, *IEEE International Conference on Robotics and Automation (ICRA)* (2022).

89. A. Rezazadeh, C. Choi, *KINet: Unsupervised Forward Models for Robotic Pushing Manipulation*, *IEEE Robotics and Automation Letters* (2023).

90. Y. Wang, *et al.*, *D^3Fields: Dynamic 3D Descriptor Fields for Zero-Shot Generalizable Robotic Manipulation*, *arXiv preprint arXiv:2309.16118* (2023).

91. W. Wang, A. S. Morgan, A. M. Dollar, G. D. Hager, *Dynamical scene representation and control with keypoint-conditioned neural radiance field*, *IEEE International Conference on Automation Science and Engineering (CASE)* (2022).

92. E. S. Spelke, *Principles of Object Perception*, *Cogn. Sci.* **14**, 29 (1990).

93. M. Janner, *et al.*, *Reasoning About Physical Interactions with Object-Oriented Prediction and Planning*, *International Conference on Learning Representations (ICLR)* (2019).

94. K. Yi, *et al.*, *CLEVRER: Collision Events for Video Representation and Reasoning*, *International Conference on Learning Representations (ICLR)* (2020).

95. H. Qi, X. Wang, D. Pathak, Y. Ma, J. Malik, *Learning Long-term Visual Dynamics with Region Proposal Interaction Networks*, *International Conference on Learning Representations (ICLR)* (2021).

96. N. Watters, *et al.*, *Visual interaction networks: Learning a physics simulator from video*, *Advances in neural information processing systems (NeurIPS)* (2017).

97. R. Veerapaneni, *et al.*, *Entity Abstraction in Visual Model-Based Reinforcement Learning*, *Conference on Robot Learning (CoRL)* (2020).

98. S. Tian, *et al.*, *Multi-Object Manipulation via Object-Centric Neural Scattering Functions*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

99. M. Chang, T. D. Ullman, A. Torralba, J. B. Tenenbaum, *A Compositional Object-Based Approach to Learning Physical Dynamics*, *International Conference on Learning Representations (ICLR)* (2017).

100. P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, K. kavukcuoglu, *Interaction networks for learning about objects, relations and physics*, *Advances in Neural Information Processing Systems (NeurIPS)* (2016).

101. J. B. Tenenbaum, V. de Silva, J. C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, *Science* **290**, 2319 (2000).

102. H. Yu, *et al.*, *Learning Object-Centric Neural Scattering Functions for Free-viewpoint Relighting and Scene Composition*, *Trans. Mach. Learn. Res.* (2023).

103. Z. Huang, X. Lin, D. Held, *Mesh-based Dynamics with Occlusion Reasoning for Cloth Manipulation*, *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July 1, 2022*, K. Hauser, D. A. Shell, S. Huang, eds. (2022).

104. S. M. LaValle, *Rapidly-exploring random trees : a new tool for path planning*, *The annual research report* (1998).

105. L. E. Kavraki, P. Svestka, J. Latombe, M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, *IEEE Trans. Robotics Autom.* (1996).

106. R. Y. Rubinstein, D. P. Kroese, *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)* (Springer-Verlag, 2004).

107. G. Williams, A. Aldrich, E. A. Theodorou, *Model Predictive Path Integral Control using Covariance Variable Importance Sampling*, *arXiv:1509.01149* (2015).

108. K. Zhang, B. Li, K. Hauser, Y. Li, *AdaptiGraph: Material-Adaptive Graph-Based Neural Dynamics for Robotic Manipulation*, *Robotics: Science and Systems* (2024).

109. R. S. Sutton, *Dyna, an Integrated Architecture for Learning, Planning, and Reacting*, *SIGART Bull.* (1991).

110. M. Janner, J. Fu, M. Zhang, S. Levine, *When to Trust Your Model: Model-Based Policy Optimization*, *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

111. D. Bauer, Z. Xu, S. Song, *DoughNet: A Visual Predictive Model for Topological Manipulation of Deformable Objects*, *European Conference on Computer Vision (ECCV)* (2024).

112. X. Lin, Y. Wang, Z. Huang, D. Held, *Learning Visible Connectivity Dynamics for Cloth Smoothing*, *Conference on Robot Learning (CoRL)* (2021).

113. H. Li, *et al.*, *See, Hear, and Feel: Smart Sensory Fusion for Robotic Manipulation*, *Conference on Robot Learning (CoRL)* (2022).

114. H. Thomas, *et al.*, *KPConv: Flexible and Deformable Convolution for Point Clouds*, *IEEE International Conference on Computer Vision (ICCV)* (2019).

115. R. Bommasani, *et al.*, *On the Opportunities and Risks of Foundation Models*, *arXiv:2108.07258* (2021).

116. D. Schmidt, M. Jiang, *Learning to Act without Actions*, *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024* (OpenReview.net, 2024).

117. Y. R. Wang, J. Duan, D. Fox, S. S. Srinivasa, *NEWTON: Are Large Language Models Capable of Physical Reasoning?*, *Findings of the Association for Computational Linguistics (EMNLP)* (2023).

118. Q. Gao, *et al.*, *Do Vision-Language Models Have Internal World Models? Towards an Atomic Evaluation*, *Findings of the Association for Computational Linguistics: ACL 2025* (2025).

119. B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, *3D Gaussian Splatting for Real-Time Radiance Field Rendering*, *ACM Trans. Graph.* (2023).

120. G. Wu, *et al.*, *4D Gaussian Splatting for Real-Time Dynamic Scene Rendering*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

121. T. Xie, *et al.*, *PhysGaussian: Physics-Integrated 3D Gaussians for Generative Dynamics*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024* (IEEE, 2024), pp. 4389–4398.

122. M. Zhang, K. Zhang, Y. Li, *Dynamic 3D Gaussian Tracking for Graph-Based Neural Dynamics Modeling*, *Conference on Robot Learning* (2024).

123. C. Cadena, *et al.*, *Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age*, *IEEE Trans. Robotics* **32**, 1309 (2016).

124. Y. LeCun, *A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27*, *Open Review* (2022).

125. C. R. Garrett, *et al.*, *Integrated Task and Motion Planning*, *Annu. Rev. Control. Robotics Auton. Syst.* **4**, 265 (2021).

126. S. Li, *et al.*, *DexDeform: Dexterous Deformable Object Manipulation with Human Demonstrations and Differentiable Physics*, *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (OpenReview.net, 2023).

127. D. Silver, *et al.*, *Mastering the game of Go without human knowledge*, *Nature* **550**, 354 (2017).

128. Y. Lee, P. Cai, D. Hsu, *MAGIC: Learning Macro-Actions for Online POMDP Planning*, *Robotics: Science and Systems* (2021).

129. H. Qi, H. Yin, Y. Du, H. Yang, *Strengthening Generative Robot Policies through Predictive World Modeling*, *CoRR* **abs/2502.00622** (2025).

130. K. Chua, R. Calandra, R. McAllister, S. Levine, *Deep reinforcement learning in a handful of trials using probabilistic dynamics models*, *Neural Information Processing Systems (NeurIPS)* (2018).

131. C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, *Weight uncertainty in neural networks*, *International Conference on International Conference on Machine Learning (ICML)* (2015).

132. A. Nagabandi, K. Konolige, S. Levine, V. Kumar, *Deep dynamics models for learning dexterous manipulation*, *Conference on Robot Learning (CoRL)* (2020).

Funding: This research is in part funded by the NSF AI-Center TILOS and Hillbot Embodied AI Fund. S. Tian was supported by NSF GRFP Grant No. DGE-1656518. C. Tan was supported by A*STAR CRF funding. Y. Li is partially supported by the Toyota Research Institute (TRI), the Sony Group Corporation, Google, Dalus AI, and the DARPA TIAMAT program (HR0011-24-9-0430). This article solely reflects the opinions and conclusions of its authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

Competing interests: Henrik I. Christensen has equity interests in Robust.AI which develops Warehouse Robots and in Parxis Solutions that develops Financial AI software. Hao Su has

equity interests in HillBot and is the CTO of Hillbot. Yunzhu Li holds equity in and serves as an advisor to SceniX Inc.