MoRE: Mixture-of-Experts for Multi-Physics Robotic World Models

Author Names Omitted for Anonymous Review. Paper-ID [1066]

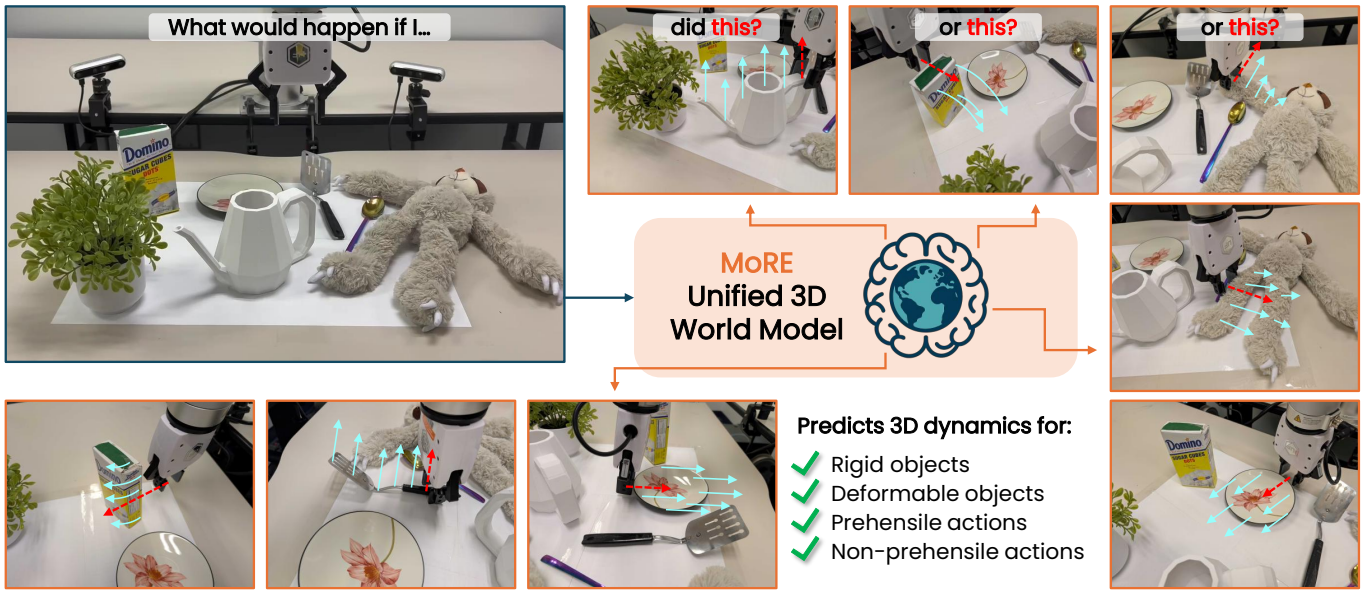


Fig. 1: **MoRE: Mixture-of-Motion Experts for Multi-Physics Robotic World Models.** MoRE predicts action-conditioned 3D scene dynamics across diverse manipulation settings by routing interactions to specialized motion experts. Operating directly in 3D point cloud space, our model captures contact-rich interactions between different tool geometries and objects under heterogeneous physical regimes, including rigid pushing and deformable manipulation in both prehensile and non-prehensile settings. Trained at scale in simulation and integrated with model predictive control, MoRE enables accurate, real-time (20 FPS) dynamics prediction and generalizes across tasks, materials, and object instances.

***Abstract*—A robotic world model should predict how objects evolve under robot actions and physical dynamics. Existing 3D world modeling approaches can only handle a narrow set of object interactions per model, failing to generalize to new objects. In this paper, we propose MoRE, a novel 3D world model that switches between different "motion experts" that each handle a different type of interaction (e.g. rigid-body vs. soft-body, prehensile vs. non-prehensile). This design allows a single model to efficiently learn a wide variety of physical dynamics, enabling generalization to unseen objects. We train a 1B-parameter version of our model on a diverse dataset of object interactions collected in IsaacSim and demonstrate that integrating our world model with Model Predictive Control (MPC) enables a range of manipulation tasks using a single model. Experiments show that our method outperforms existing approaches on both rigid and deformable dynamics prediction benchmarks and real-world tasks, highlighting its effectiveness for robotic manipulation.**

I. Introduction

Robots that operate in the physical world must predict how the environment evolves under their actions. A robotic world model [65, 67, 25, 20, 42, 19, 57] predicts future scene evolution conditioned on robot controls, which is fundamental for planning, control, and long-horizon decision making. This is especially crucial in contact-rich real-world manipulation [49, 55, 35, 61, 60]. For instance, a house cleaning robot must anticipate cloth and object motion; a cooking robot must predict material deformation under tools.

Building such a dynamics model is hard. A useful robotic world model must be accurate, physical-plausible, 3D-aware, and interaction-grounded. It must handle diverse physical regimes, from rigid objects to deformable materials. It must also support diverse manipulation modes, including both prehensile and non-prehensile interaction, with grippers and a wide range of rigid or soft tools. In short, we need models that are geometric-grounded, multi-physics, multi-tool, and contact-rich.

Existing approaches address only part of this problem. Physics-based methods [28, 58, 15] leverage differentiable simulators and support long-horizon rollouts, but they often require heavy system identification and expensive optimiza-

tion. Learning-based dynamics models [63, 10, 33, 36, 39, 59, 5, 47, 50, 33, 61] are typically narrow: they focus on a single material type, tool, or task, and generalize poorly. Generative video world models [19, 22, 67] scale well, but they lack accurate 3D structure and action grounding, which limits their utility for real robotic execution. A universal robotic world model that is fully 3D, multi-physics, multi-tool, and precise remains out of reach.

We aim to close this gap. We propose MoRE, a scalable robotic world model that takes a short history of object–tool interaction and predicts future 3D point trajectories conditioned on the current action. Our core philosophy is simple: combine scaling with physics-informed architecture design. We scale data aggressively across simulation and real settings, spanning diverse object geometries, materials, contact types, tasks, and tools. This diversity is critical for generalization. At the same time, we argue that robotics dynamics is fundamentally heterogeneous, and should not be forced into a single monolithic predictor. Inspired by interaction structure [5, 33, 60, 61, 46] in robotics and the success of Mixture-of-Experts [48, 16], we introduce a Mixture-of-Motion Experts architecture. We decompose dynamics into functional experts aligned with physical regimes and interaction types (e.g., rigid vs. non-rigid, prehensile vs. non-prehensile). All experts share a common shape-and-motion encoder, but each expert is grounded by appropriate constraints and inductive biases (e.g., rigidity preservation). A learned gating module routes each interaction to the right expert based on short motion history, effectively inferring material and interaction mode from observation.

We evaluate MoRE in both simulation and real-world manipulation, covering pushing and grasping with rigid and deformable objects. Trained on a large-scale interaction dataset, our 1B-parameter model runs in real time (20 FPS) and integrates naturally with tuning-free MPC for closed-loop control. Across dynamics benchmarks and real robot tasks, MoRE consistently outperforms prior methods.

In summary, we make three contributions. First, we propose a 3D, point-based robotic world model for contact-rich dynamics prediction. Second, we introduce MoRE, a Mixture-of-Motion Experts framework designed explicitly for heterogeneous physics. Third, we demonstrate strong real-time performance and closed-loop control on both simulated and real-world robotic manipulation tasks.

II. Related Work

Existing robotic dynamic models can be broadly divided into three categories: physics-based simulation methods, learning-based dynamics models, and foundation robotic world models. A comparison between our approach and representative prior work across these categories is shown in Table I.

A. Physics-Based Dynamic Models

Physics-based dynamics models [56, 45, 15] predict manipulation outcomes by explicitly simulating physical interactions. For rigid bodies [38, 17, 30], common formulations use state-based dynamics on the $\mathrm{SE}(3)$ manifold with contact constraints, implemented in rigid-body simulators [2, 3, 56, 12, 11]. For deformable and soft objects, simulators typically rely on continuum or particle-based representations, such as MPM [58, 7, 13, 52, 23, 54], PBD [41, 37, 1], or spring-mass systems [64, 28], to capture material-dependent deformation. Recent hybrid pipelines [58, 62, 8, 31] combine reconstruction with physics-informed simulation to build digital twins for future prediction. These approaches provide strong physical priors and support long-horizon rollouts when parameters are accurate, but often depend on system identification and expensive test-time optimization to match observations [28, 58]. Moreover, they typically require complete state and detailed geometry as input, making them suffer under partial observations, sensor noise, or unknown materials. In contrast, our method is feed-forward and physically grounded, avoiding test-time optimization and explicit parameter identification.

TABLE I: **Comparison of robotic manipulation methods** across key capabilities: feed-forward inference and unified modeling of multi-physics dynamics (different material properties of manipulated objects, e.g., rigid vs. deformable) and multi-tool interactions (different end-effector geometries and contact interfaces), as well as 3D-grounded geometry. ParticleFormer shows partial multi-tool support but is trained per scene. PointWorld relies on implicit dynamics while we explicitly routes interactions to physics-aligned motion experts.

Method	**Feed-forward**	**Multi-phys.**	**Multi-tool**	**3D**
PhysTwin [28]	✗	✗	✗	✓
PhysGaussian [58]	✗	✗	✗	✓
AdaptiGraph [60]	✗	✗	✗	✓
ParticleFormer [24]	✓	✓	✓†	✓
Points2Plans [27]	✓	✗	✗	✓
RoboDreamer [67]	✓	✗	✗	✗
DINO-WM [65]	✓	✓	✓	✗
PointWorld [25]	✓	✓	✓	✓
Ours	✓	✓	✓	✓

B. Learning-Based Dynamic Models

Learning-based dynamics models [63, 10, 33, 36, 39, 59] bypass explicit simulation by learning action-conditioned state transitions from data. Early work represents objects and interactions using graphs [33, 44], particles [61, 18, 33], or GNNs [5, 47, 50], while more recent approaches adopt transformer-style architectures to improve capacity and scalability for contact-rich interactions. These models enable efficient inference and strong performance in specific regimes, operating directly from observations. However, those models remain *narrow*: they are trained on category or task-specific datasets and implicitly assume fixed materials, tools, or interaction modes. This often leads to overfitting and limited generalization ability. Our model is also learning-based, but is explicitly designed to support diverse materials, physical properties, object geometries, and action-conditioned interactions within a unified framework.

C. Foundation Robotic World Models

Motivated by the success of large visual foundation models [42, 4, 43], recent work [19, 46, 21, 40, 22] build robotic world models on top of such models. For example, DINO-WM [65] models dynamics and performs planning in a DINOv2 [43] feature space, while RoboDreamer [67] uses video diffusion models [29] to synthesize future videos for planning. However, these approaches are not explicitly grounded in 3D geometry or contact interactions, which can limit physical fidelity. Another line of work focuses on 3D foundation world models that predict scene evolution directly in geometric space [25]. The concurrent work PointWorld [25] predicts action-conditioned 3D point flow from scene point clouds and robot motion with implicit representations. In contrast, our approach explicitly decomposes dynamics by interaction type using learned gating, routing interactions to specialized motion experts aligned with physical regimes and enforcing regime-specific constraints, enabling more physically faithful prediction across heterogeneous manipulation scenarios.

III. Method

A. Problem Formulation

We model a robotic *3D world model* as a dynamical system that predicts future scene evolution under robot actions and physical interactions. Let $\mathbf{s}_t \in \mathcal{S}$ denote the *current* scene state and $\mathbf{a}_t \in \mathcal{A}$ the *current* robot action, where $\mathcal{S}$ and $\mathcal{A}$ are geometry-based point-cloud spaces defined below. Given the current state–action pair $(\mathbf{s}_t, \mathbf{a}_t)$ together with a finite history of past states and actions $\{\mathbf{s}_{t-H':t-1}, \mathbf{a}_{t-H':t-1}\}$, the goal is to predict future scene states over a horizon of H steps.

We parameterize the world model as

$$\mathbf{s}_{t+1:t+H} = f_\theta\big(\mathbf{s}_t,\, \mathbf{a}_t,\, \mathbf{s}_{t-H':t-1},\, \mathbf{a}_{t-H':t-1}\big). \tag{1}$$

We represent each scene state as a 3D point cloud, $\mathbf{s}_t \in \mathcal{S} := \mathbb{R}^{N_S \times 3}$ with N_S being the number of scene points. Crucially, we design the action space $\mathcal{A}$ in a geometry-centric manner: unlike most prior work that conditions dynamics on joint commands, end-effector poses, or sparse keypoints [32, 51, 14], we represent each action by the full 3D state of the *tool* as a point cloud, $\mathbf{a}_t \in \mathcal{A} := \mathbb{R}^{N_A \times 3}$. Specifically, low-level robot controls are mapped via forward kinematics to $\mathbf{a}_t$, where the tool is any directly actuated geometry by the embodiment (e.g., grippers, hands, cookware, or daily tools), as long as its state can be computed via forward kinematics. This representation decouples the world model from embodiment-specific kinematics, enabling cross-embodiment transfer and multi-tool generalization across diverse manipulation tasks. Conditioned on the current and historical scene–action point clouds, the model predicts future scene point clouds $\mathbf{s}_{t+1:t+H}$, capturing rigid-body motion, deformation, and contact-induced dynamics.

B. Mixture-of-Experts Dynamic Modeling

Real-world physical interactions span diverse regimes, from rigid to non-rigid dynamics and from prehensile to non-prehensile manipulation. These regimes impose distinct kinematic and kinetic constraints, making a single monolithic transformer difficult to learn and optimize for uniformly accurate prediction. Yet, we observe that the interaction regime is often straightforward to infer from a short history of states and actions. Motivated by this, we adopt a Mixture-of-Experts architecture that identifies motion patterns and routes to specialized dynamics decoders for generic multi-physics modeling.

Given the current and historical scene states and actions represented as point clouds, we encode them using a scene encoder and an action encoder to obtain point *embeddings*. The embeddings are fused into a unified representation $\mathbf{z}_t$, which serves as a shared conditioning for both the gating function and all expert dynamics models. We predict a one-hot routing vector via a gating function $g(\cdot)$:

$$\mathbf{g}_t = g_\theta(\mathbf{z}_t);\ \text{where}\ \ \mathbf{g}_t \in \{0,1\}^K, \sum_{k=1}^{K} g_t^{(k)} = 1, \tag{2}$$

where K is the number of experts. This hard routing encourages clear expert specialization and reduces interference across heterogeneous physical regimes.

Each expert dynamics model f_k takes the same shared embedding $\mathbf{z}_t$ as input and predicts the next H-step scene states. The overall MoE prediction is

$$\mathbf{s}_{t+1:t+H} = \sum_{k=1}^{K} g_t^{(k)}\, f_k(\mathbf{z}_t), \tag{3}$$

which is equivalent to activating only the selected expert under one-hot routing. The overall architecture is shown in Fig. 2.

C. Physics-Aware Expert Design

A key advantage of our MoE formulation is that expert routing is physically meaningful. Each expert corresponds to a distinct interaction regime with different kinematic and dynamic constraints, enabling not only clearer specialization but also regime-specific parameterizations and training objectives. In our instantiation, we use three experts: (i) *rigid non-prehensile*, (ii) *deformable non-prehensile*, and (iii) *deformable prehensile*. We omit rigid prehensile interactions, since rigid grasping typically induces limited physics uncertainty when slip is negligible, and can be well modeled by existing world models.

a) Rigid Non-Prehensile Expert.: For rigid objects, motion is fully characterized by a global 6-DoF rigid transform $(\mathbf{R}_\tau, \mathbf{t}_\tau) \in \mathrm{SE}(3)$. Accordingly, the rigid expert predicts a sequence of transforms via an internal head

$$(\boldsymbol{\omega}_{t+1:t+H}, \mathbf{t}_{t+1:t+H}) = h_{\text{rigid}}(\mathbf{z}_t), \tag{4}$$

where $\boldsymbol{\omega}_t \in \mathbb{R}^6$ is the 6D continuous rotation representation (converted to $\mathbf{R}_t \in \mathrm{SO}(3)$) and $\mathbf{t}_t \in \mathbb{R}^3$ is translation. The expert then outputs future scene point clouds by rigid warping

$$f_{\text{rigid}}(\mathbf{z}_t; \tau) = \mathbf{R}_{t+1:t+H} \cdot \mathbf{s}_t + \mathbf{t}_{t+1:t+H} \tag{5}$$

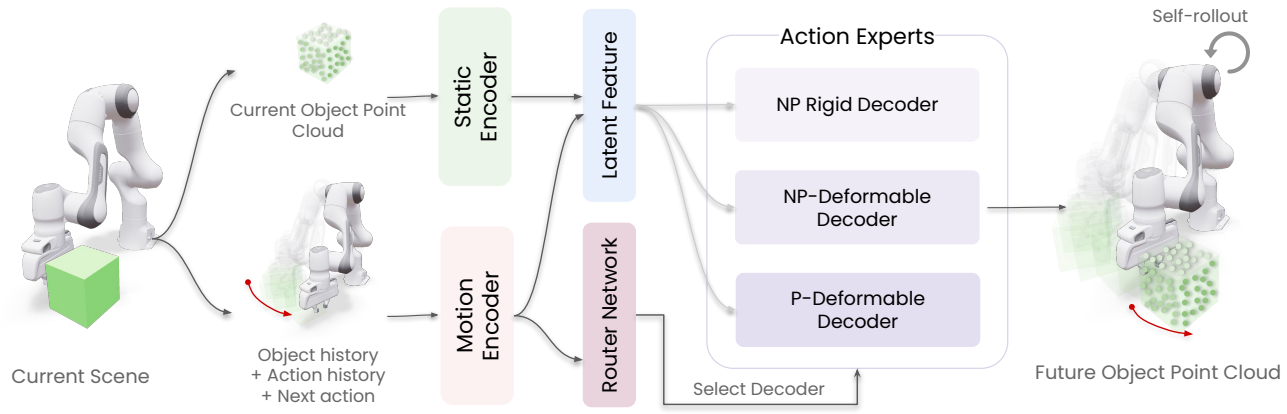


Fig. 2: **Overview of MoRE**. Given the current and historical object states and actions, represented as point clouds, we first encode the scene and actor inputs into point tokens using two encoders. An explicit gating network then selects among specialized motion experts, each tailored to a distinct physical regime, to predict the next object state in 3D. P and NP denote "prehensible" and "non-prehensible", respectively.

This parameterization enforces rigidity and global coherence, reducing the hypothesis space and improving stability compared to free-form point decoding.

b) Deformable Dynamics Experts.: Deformable objects require free-form geometry updates beyond global rigid warping. We therefore instantiate *two* deformable experts with identical architecture but separate parameters and training: one specialized for prehensile interactions and one for non-prehensile interactions. The prehensile expert specializes in force-induced shape changes under grasping (e.g., squeezing and stretching), whereas the non-prehensile expert focuses on contact-induced deformations from pushing and sliding interactions. For simplicity, we omit explicit expert indices in the notation.

Given the shared embedding $\mathbf{z}_t$, the selected deformable expert predicts point-wise displacements via an internal head

$$\Delta \mathbf{s}_{t+1:t+H} = h_{\text{def}}(\mathbf{z}_t), \tag{6}$$

and outputs future scene states by applying the displacement

$$f_{\text{def}}(\mathbf{z}_t) = \mathbf{s}_{t:t+H-1} + \Delta \mathbf{s}_{t+1:t+H}. \tag{7}$$

D. Training and Inference

a) Training: During training, we adopt a self-rolling strategy [26] to match inference-time behavior and mitigate distribution shift. Starting from the current scene state $\mathbf{s}_t$ (and the history buffer), the model predicts the next-step scene $\hat{\mathbf{s}}_{t+1}$. We then feed $\hat{\mathbf{s}}_{t+1}$ back as input to predict subsequent steps, rolling out autoregressively over a horizon of H steps. Following prior work, gradients are detached through the rolled-in predictions, and losses are computed over the entire rollout horizon.

We optimize a combined objective consisting of a geometric reconstruction loss and an expert routing loss:

$$\mathcal{L} = \sum_{\ell=1}^{H} \left\| \mathbf{s}_{t+\ell} - \mathbf{s}^{\text{gt}}_{t+\ell} \right\|_2^2 + \lambda \sum_{\ell=1}^{H} \text{CE}\left(\mathbf{g}_{t+\ell},\, \mathbf{g}^{\text{gt}}_{t+\ell}\right), \tag{8}$$

where $\mathbf{s}^{\text{gt}}$ denotes ground-truth scene point clouds, $\hat{\mathbf{g}}$ is the predicted one-hot routing distribution from the gating function $g(\cdot)$, and $\mathbf{g}^{\text{gt}}$ is the supervision signal for expert selection.

We first warm up the gating network and experts separately in simulation, where expert labels (e.g., material / interaction regime) are available. We then perform joint fine-tuning of the full MoE world model on real data.

b) Inference: At inference time, the model performs fully autoregressive rollout. Given the observed initial scene and a history buffer initialized from past states/actions, we iteratively predict $\mathbf{s}_{t+1}, \mathbf{s}_{t+2}, \ldots$. After each step, the predicted state is appended to the history buffer and used as input for the next prediction. The procedure repeats until reaching the desired prediction horizon.

E. Planning with World Model

We use the learned world model as the predictive dynamics in an Model Predictive Control (MPC) loop. At each control step, given the current observed state s_t (scene point cloud) and history, we sample a set of candidate action sequences $\{\mathbf{a}^{(j)}_{t:t+H-1}\}_{j=1}^{J}$ over a planning horizon H and roll out the world model to obtain predicted trajectories $\{\mathbf{s}^{(j)}_{t+1:t+H}\}$. We evaluate each trajectory with a task-specific cost that measures progress toward a goal state s_{goal} (e.g., distance between the predicted terminal scene and the goal), and select the lowest-cost plan:

$$j^\star = \arg\min_j \; \mathcal{C}\left(\mathbf{s}^{(j)}_{t+1:t+H}, s_{\text{goal}}\right). \tag{9}$$

We then execute only the first action $a_t^{(j^\star)}$, observe the next state, and re-plan at the next step. For obstacle present (e.g. multi-object manipulation), we add additional cost for collision avoidance. Other MPC details (action sampling, costs, and constraints) follow standard practice and are provided in the following section.

TABLE II: **Comparison of dynamic modeling methods** across interaction types and physics. Results are reported using geometric accuracy (MSE, CD) and material-consistent structural metrics (rigidity or Laplacian). Neural dynamics models (e.g., AdaptiGraph, ParticleFormer) and single-regime experts perform well in narrow settings but struggle to generalize across contact-rich interactions, while our Mixture-of-Experts model achieves consistently strong performance across all regimes.

	Rigid-body Non-Prehensile			**Deformable Non-Prehensile**			**Deformable Prehensile**		
Method	**RMSE ↓**	**CD ↓**	**Rigidity ↓**	**RMSE ↓**	**CD ↓**	**Laplacian ↓**	**RMSE ↓**	**CD ↓**	**Laplacian ↓**
AdaptiGraph	0.028	0.084	0.0001	0.031	0.052	0.0068	0.047	0.143	0.0092
ParticleFormer	0.054	0.082	0.0005	0.061	0.079	0.0072	0.058	0.075	0.0058
Rigid-Expert	**0.021**	**0.048**	**0.0**	–	–	–	–	–	–
Deform-NP-Expert	–	–	–	**0.018**	0.032	0.0027	–	–	–
Deform-P-Expert	–	–	–	–	–	–	0.029	**0.038**	0.0068
Ours (MoE)	0.022	**0.048**	**0.0**	0.024	**0.027**	**0.0020**	**0.026**	0.039	**0.0050**

TABLE III: **Ablation of MoRE** under different interaction types and physical regimes. We evaluate the impact of expert specialization, routing strategy, and self-rollout training using geometric and structural metrics. Performance degrades when removing expert specialization, explicit routing, or self-rollout training, highlighting the importance of each component for robust dynamics prediction.

	Rigid-body Non-Prehensile			**Deformable Non-Prehensile**			**Deformable Prehensile**		
Method	**RMSE ↓**	**CD ↓**	**Rigidity ↓**	**RMSE ↓**	**CD ↓**	**Laplacian ↓**	**RMSE ↓**	**CD ↓**	**Laplacian ↓**
Single Expert	0.030	0.054	0.0001	0.028	0.038	0.0038	0.037	0.047	0.0061
Implicit Router	0.026	0.051	**0.0**	0.032	0.033	0.0032	0.028	0.040	0.0054
Implicit MoE	0.045	0.070	0.0002	0.026	0.031	0.0032	0.032	**0.039**	0.0071
w/o Self-Rollout	0.031	0.057	**0.0**	0.035	0.039	0.0056	0.047	0.060	0.0058
Ours (Full)	**0.022**	**0.048**	**0.0**	**0.024**	**0.027**	**0.0020**	**0.026**	**0.039**	**0.0050**

IV. Experiments

A. Implementation Details

a) Dataset Curation: We curate a mixed simulation–real dataset covering diverse physical regimes and interactions. In simulation, we include 300 rigid objects and 300 deformable objects from [9, 53], as well as 30 different tools from [34]. For non-prehensile pushing, actions are synthesized by randomly sampling contact points on the object surface and computing surface normals to define push directions; object properties are randomized per episode, including mass and friction for rigid objects and Young's modulus for deformable objects, to encourage robustness and generalization. Rigid and deformable objects are both manipulated via randomized pushing and reaching interactions in simulation. For prehensile deformable manipulation, we use real-world RGB-D data from prior work [61]. The final dataset combines simulated and real trajectories across tools, objects, and interaction regimes.

b) Model Architecture: We use a fixed-size point representation with 512 object points and 100 end-effector (tool) points, and a temporal window consisting of 8 history steps and a 3-step prediction horizon. At each step, the current object point cloud is encoded by a frozen Uni3D [66] backbone to extract per-point static geometry features, while historical object points together with past and current tool points are embedded using lightweight point-wise projections with coordinate positional encodings to form dynamic tokens. These tokens are processed by a motion encoder built on a Uni3D backbone, and the resulting motion features are fused with static geometry through a cross-attention module, where static object features act as queries and motion features provide keys and values, producing geometry-aware motion representations for each object point. The fused features are not only used by expert heads, which predict either a global rigid motion or per-point deformations, but also fed into an explicit gating network enabling a hard routing decision. Training follows a self-rolling strategy, predicting one step at a time and feeding the prediction back as input, repeating this process three times to match inference-time autoregressive rollout.

B. Tasks and Evaluation Setup

a) Data: We train and evaluate our model on a large-scale collection of dynamic point cloud rollouts spanning diverse physical regimes. The dataset includes rigid-body and deformable objects, prehensile and non-prehensile interactions, as well as inertial dynamics without external forces. Most rollouts are generated in the IsaacSim environment, covering tool-based pushing, manipulation, and grasping scenarios. To improve real-world generalization, we additionally collect real-world recordings of deformable object interactions. The final training set combines simulated and real data, and we construct a unified benchmark that evaluates all methods consistently across rigid and deformable, prehensile and non-prehensile regimes.

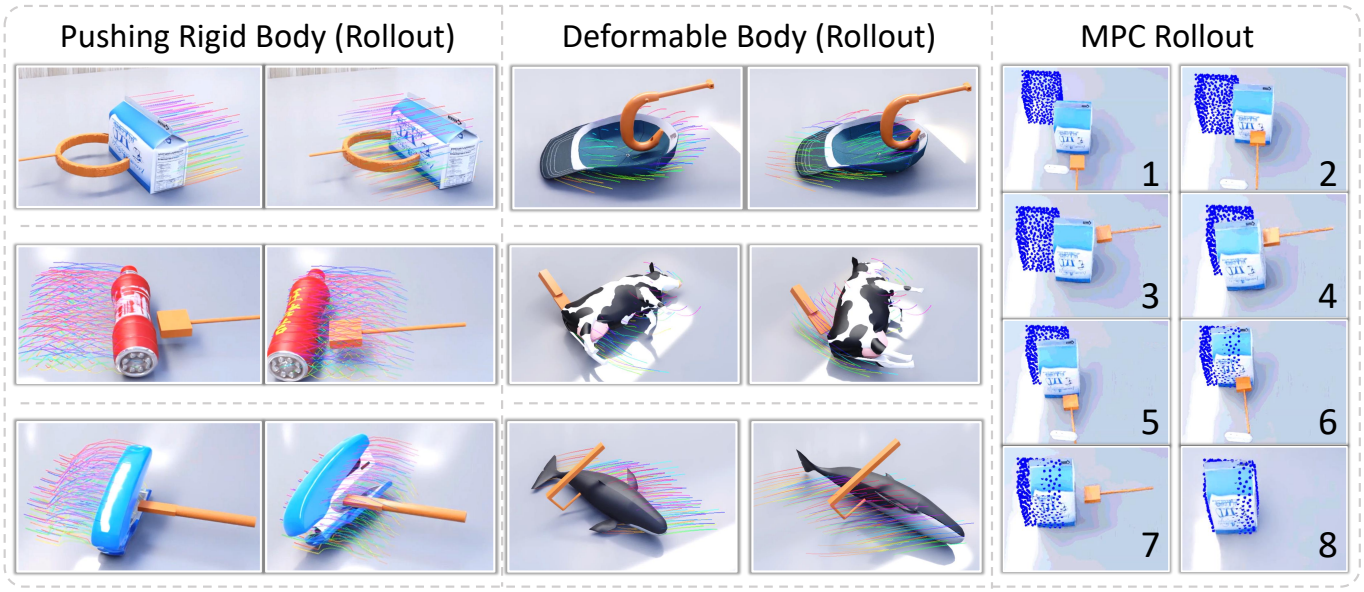


Fig. 3: **Dynamic prediction and planning in simulation. Left:** rigid non-prehensile pushing, where the model accurately simulates the rolling behavior of bottles under end-effector contact. **Middle:** deformable object interactions under non-prehensile and prehensile contact, exhibiting both local deformations and global object motion. **Right:** MPC rollouts in simulation, visualizing self-rolled trajectories over time. Across all cases, the model preserves rigidity and captures contact-induced deformations across tools and interaction regimes.

b) Tasks: We evaluate both dynamic modeling accuracy and downstream control performance across three representative manipulation regimes: rigid-body non-prehensile pushing, deformable non-prehensile manipulation, and deformable prehensile interaction. Rigid-body tasks involve tool-based interaction with everyday objects such as boxes and containers, emphasizing object coherence and contact consistency. Deformable manipulation tasks involve soft objects such as toys, ropes, and cloth-like materials, requiring accurate modeling of local surface deformation. Deformable prehensile tasks further introduce grasping and sustained contact, posing additional challenges due to complex contact dynamics and large non-rigid shape changes.

c) Metrics and Evaluation Setup: We evaluate dynamic prediction quality using both geometric accuracy and material-consistent structural metrics, as summarized in Table II. Geometric accuracy is measured by mean squared error (MSE) and Chamfer Distance (CD) between predicted and ground-truth point clouds. For rigid-body interactions, we report a rigidity error computed by fitting an optimal rigid transformation between predicted and ground-truth point clouds using ICP and measuring the residual alignment error, where lower values indicate better rigidity preservation. For deformable objects, we report a Laplacian error that measures differences in Laplacian coordinates between predicted and ground-truth point clouds, with lower values corresponding to smoother and more physically plausible deformations. All evaluations are conducted in both simulation and real-world settings using identical protocols.

C. Dynamic Modeling across Interactions and Physics

Table II compares dynamic modeling performance across three representative manipulation regimes: rigid-body non-prehensile interaction, deformable non-prehensile interaction, and deformable prehensile interaction.

We compare against different baseline models with different inductive biases. AdaptiGraph [60] and ParticleFormer [24] are neural dynamics models designed primarily for deformable object modeling. Rigid-Expert and Deformable-Expert correspond to category-specific dynamics experts with the same architecture trained independently for rigid-body and deformable interactions, respectively.

Across all regimes, existing methods perform well only within their targeted categories and struggle to generalize across interaction types and physical properties. In contrast, our mixture-of-experts (MoE) model consistently achieves strong performance across all settings. By combining explicit expert specialization with a shared representation and history-conditioned routing, our model effectively captures heterogeneous manipulation dynamics, yielding improved accuracy and more physically consistent predictions across rigid and deformable, prehensile and non-prehensile interactions. The visualizations are shown in Fig. 3.

D. Ablation Study on Motion Experts and Routing

Table III presents an ablation study evaluating the impact of architectural and training design choices across interaction types and physical regimes. Training a single expert to model all physics leads to degraded performance, indicating that a unified dynamics model struggles to capture both rigid and

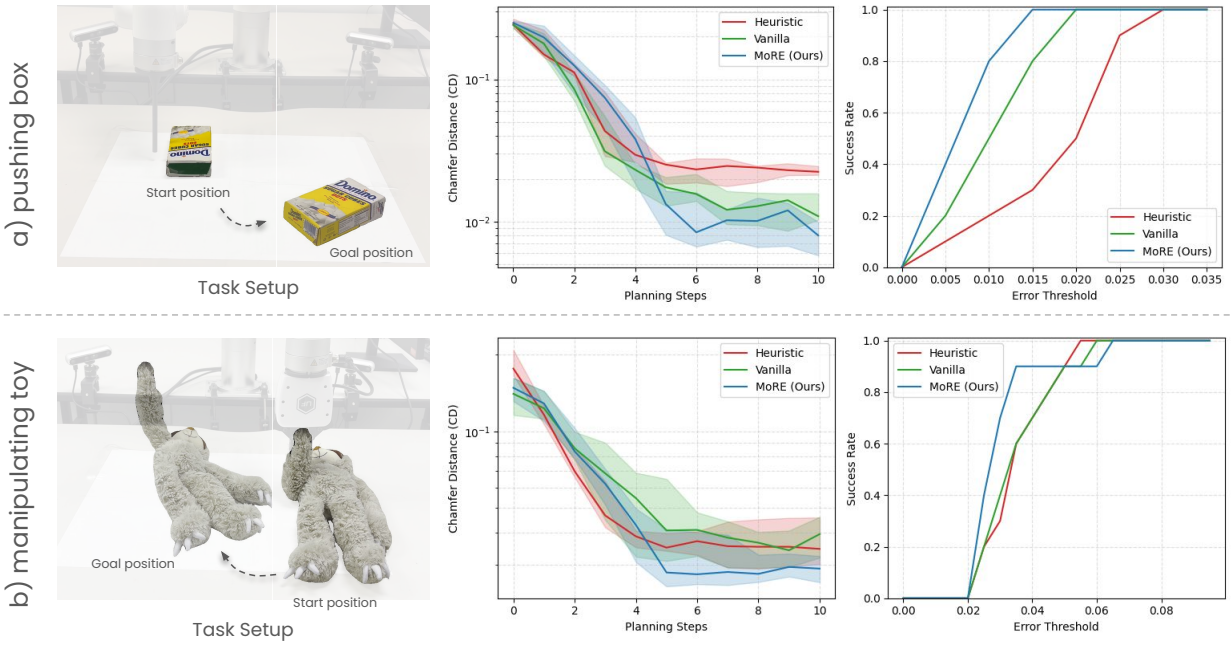


Fig. 4: **Model Predictive Control (MPC) with learned world models. Left:** Action-conditioned prediction error measured by Chamfer Distance (CD) as a function of planning steps, showing how prediction quality evolves during MPC rollouts. **Right:** Task success rate as a function of error threshold, measuring robustness of planning under model inaccuracies. Top row corresponds to a rigid pushing task (pushing box), while the bottom row evaluates a contact-rich manipulation task (manipulating toy). Across both tasks, MoRE consistently achieves lower prediction error and higher success rates compared to heuristic and vanilla baselines, demonstrating the benefit of contact-aware, multi-physics world modeling.

deformable behaviors. Implicit routing, which softly combines expert outputs without supervised, binary gating, further reduces geometric accuracy and physical consistency, highlighting the importance of explicit expert selection. Implicit MoE embeds expert specialization within the transformer using a shared, displacement-based output representation, rather than employing explicit motion experts with distinct roles. This design consistently underperforms across all regimes, suggesting that implicit capacity allocation alone is insufficient to capture heterogeneous dynamics without explicit expert separation. Finally, removing self-rollout training and relying solely on teacher forcing results in less stable predictions, particularly for contact-rich deformable interactions. Overall, these results demonstrate that explicit expert specialization, supervised routing, and rollout-based training are all critical for accurate and physically consistent dynamic modeling.

E. MPC Performance with Learned World Models

a) Setup: We evaluate downstream control performance using model predictive control (MPC) in real-world robot experiments. Evaluations cover box pushing for rigid object manipulation and toy sloth pose-matching for deformable object manipulation. The goals are defined as target object point clouds. We use bidirectional Chamfer distance between the predicted and target object point cloud as our cost function. Success rate is defined as the fraction of trials in which MPC drives the object to the target within a predefined error threshold. We perform MPC in a multi-view RGB-D setup with four calibrated RGB-D cameras observing the scene. At each control step, we use SAM3 [6] to segment the target object in each camera view and back-project the masked depth maps into 3D. The resulting partial point clouds are transformed into a common world frame using known camera extrinsics and merged into a single object point cloud, which serves as the state input to the world model for prediction and planning. We consider two baselines: *Heuristic MPC*, which predicts that all points move exactly the same as the end-effector's translation; *Vanilla**, a single Transformer-based world model without motion experts or routing. For box pushing action sampling, we sample 16 candidate pushing points on the object and then sample 4 random directions for each point. For deformable toy manipulation, we start the MPC after the toy has been grasped, and sample 128 displacements uniformly randomly within a 14cm cube.

b) Results: As shown in Fig. 4, our method consistently achieves higher success rates and lower Chamfer Distance loss compared to baselines. Heuristic MPC struggles to generalize across tasks due to hand-designed, task-specific objectives, while Vanilla* suffers from limited capacity to model heterogeneous physical dynamics. In contrast, our physics-aware, expert-based 3D world model enables more accurate long-horizon prediction and efficient MPC, resulting in robust performance for both rigid and deformable manipulation and reliable transfer from simulation to real-world robotic control. Real world dynamics predictions are shown in Fig. 5.

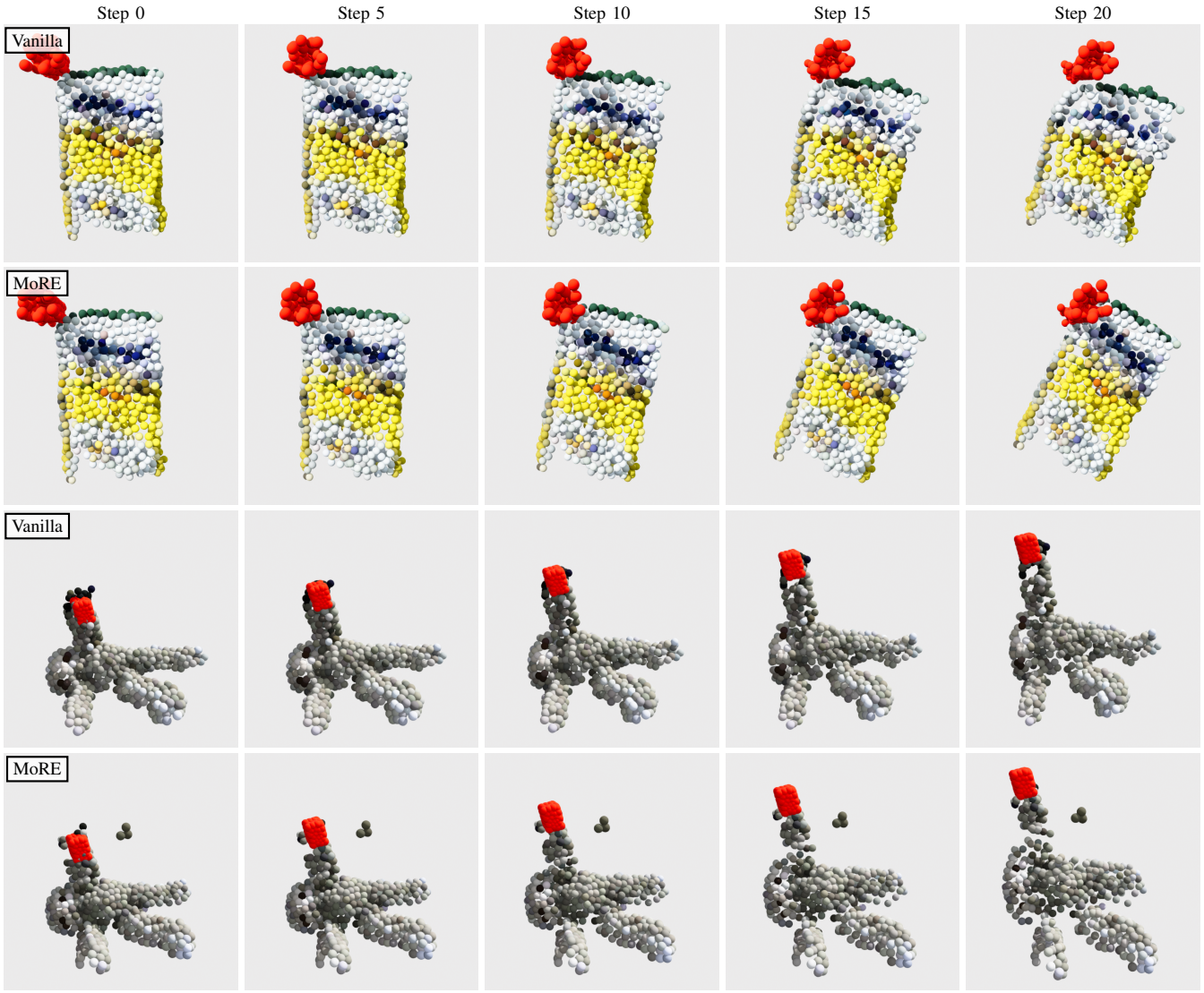


Fig. 5: **Real world dynamics predictions.** We visualize predictions of point motion on real world data. The red points are the end-effector (action) points. The Vanilla model struggles to treat the box as rigid and the toy sloth as soft. In contrast, our model correctly handles the different cases.

V. Conclusion

We introduced MoRE, a 3D-aware robotic world model for predicting interaction dynamics across heterogeneous physical regimes. By modeling scenes and actions directly in point cloud space and decomposing dynamics into specialized motion experts, MoRE enables contact-rich reasoning and robust modeling of both rigid and deformable interactions. Experiments in simulation and the real world show that MoRE achieves accurate long-horizon prediction and real-time control when integrated with MPC, consistently outperforming prior image-space and single-model baselines. We believe MoRE is a step toward scalable, general-purpose robotic world models that can support diverse manipulation tasks.

References

[1] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Tobias Pfaff, Cheston Tan, Henrik I Christensen, Hao Su, Jiajun Wu, and Yunzhu Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10(106):eadt1497, 2025.

[2] David Baraff. An introduction to physically based modeling: rigid body simulation i—unconstrained rigid body dynamics. *SIGGRAPH course notes*, 82, 1997.

[3] David Baraff. Physically based modeling: Rigid body simulation. *Siggraph Course Notes, Acm Siggraph*, 2(1): 2–1, 2001.

[4] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for

learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

[5] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.

[6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[7] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6178–6189, 2025.

[8] Hsiao-yu Chen, Edith Tretschk, Tuur Stuyck, Petr Kadlecek, Ladislav Kavan, Etienne Vouga, and Christoph Lassner. Virtual elastic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15827–15837, 2022.

[9] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025. URL https://arxiv.org/abs/2506.18088.

[10] Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos. *arXiv preprint arXiv:2205.01089*, 2022.

[11] Jacques JF Commandeur and Siem Jan Koopman. *An introduction to state space time series analysis*. Oxford university press, 2007.

[12] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[13] Alban De Vaucorbeil, Vinh Phu Nguyen, Sina Sinaie, and Jian Ying Wu. Material point method after 25 years: Theory, implementation, and applications. *Advances in applied mechanics*, 53:185–398, 2020.

[14] Norman Di Palo and Edward Johns. Keypoint action tokens enable in-context imitation learning in robotics. *arXiv preprint arXiv:2403.19578*, 2024.

[15] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (ToG)*, 41(2):1–21, 2021.

[16] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[17] Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M Kaufman, and Daniele Panozzo. Intersection-free rigid body dynamics. *ACM Transactions on Graphics*, 40(4), 2021.

[18] Shanyan Guan, Huayu Deng, Yunbo Wang, and Xiaokang Yang. Neurofluid: Fluid dynamics grounding with particle-driven neural radiance fields. In *International conference on machine learning*, pages 7919–7929. PMLR, 2022.

[19] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

[20] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

[21] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[22] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[23] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[24] 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. *arXiv preprint arXiv:2506.23126*, 2025.

[25] 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. *arXiv preprint arXiv:2601.03782*, 2026.

[26] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[27] Yixuan Huang, Christopher Agia, Jimmy Wu, Tucker Hermans, and Jeannette Bohg. Points2plans: From point clouds to long-horizon plans with composable relational

dynamics. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1208–1216. IEEE, 2025.

[28] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *arXiv preprint arXiv:2503.17973*, 2025.

[29] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

[30] Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials. *arXiv preprint arXiv:2201.10022*, 2022.

[31] Simon Le Cleac'h, Hong-Xing Yu, Michelle Guo, Taylor Howell, Ruohan Gao, Jiajun Wu, Zachary Manchester, and Mac Schwager. Differentiable physics simulation of dynamics-augmented neural objects. *IEEE Robotics and Automation Letters*, 8(5):2780–2787, 2023.

[32] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[33] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.

[34] Chunru Lin, Haotian Yuan, Yian Wang, Xiaowen Qiu, Tsun-Hsuan Wang, Minghao Guo, Bohan Wang, Yashraj Narang, Dieter Fox, and Chuang Gan. Robotsmith: Generative robotic tool design for acquisition of complex manipulation skills. *arXiv preprint arXiv:2506.14763*, 2025.

[35] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 432–448. PMLR, 2021.

[36] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022.

[37] Fei Liu, Entong Su, Jingpei Lu, Mingen Li, and Michael C Yip. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. *IEEE Robotics and Automation Letters*, 8(7):3964–3971, 2023.

[38] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pages 360–378. Springer, 2024.

[39] Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. Learning neural constitutive laws from motion observations for generalizable pde dynamics. In *International Conference on Machine Learning*, pages 23279–23300. PMLR, 2023.

[40] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023.

[41] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2): 109–118, 2007.

[42] OpenAI. Video generation models as world simulators, 2024.

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[44] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

[45] Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Differentiable simulation of soft multi-body systems. *Advances in Neural Information Processing Systems*, 34:17123–17135, 2021.

[46] Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

[47] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[48] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[49] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023.

[50] Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research*, 43(4): 533–549, 2024.

[51] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[52] Juan C Simo and Thomas JR Hughes. *Computational inelasticity*. Springer, 1998.

[53] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit

shape bias. 2021.

[54] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

[55] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM siggraph 2007 papers*, pages 80–es. 2007.

[56] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[57] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[58] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024.

[59] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. *arXiv preprint arXiv:1906.03853*, 2019.

[60] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. *arXiv preprint arXiv:2407.07889*, 2024.

[61] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. *arXiv preprint arXiv:2506.15680*, 2025.

[62] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*. Springer, 2024.

[63] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

[64] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In *European Conference on Computer Vision*, pages 407–423. Springer, 2024.

[65] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[66] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[67] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Supplementary Material – MoRE: Mixture-of-Experts for Multi-Physics Robotic World Models

Author Names Omitted for Anonymous Review. Paper-ID [1066]

The supplementary material provides implementation details of dataset curation, model architecture, MPC pipeline, and more qualitative results of dynamic rollout prediction and real-world results.

I. Implementation Details

A. Dataset Curation

Our training dataset is constructed by mixing simulated and real-world data, with a fixed mixture ratio of 1:1 throughout training. The simulated portion is collected using Isaac Lab [4], leveraging its large-scale parallel simulation capability to efficiently generate diverse and controllable object–tool interaction trajectories.

a) Assets: The simulation dataset includes three categories of assets: rigid objects, deformable objects, and tools. The rigid-object set contains approximately 300 objects spanning 130 categories, sourced from RoboTwin [2]. The deformable-object set includes around 300 objects across 100 categories, adapted from [5]. The tool set consists of approximately 30 objects from RoboSmith [3], functionally categorized into pushing, holding, lifting, and flattening tools, covering a broad range of interaction primitives.

b) Curation pipeline: Interaction data collected in simulation focus on non-prehensile manipulation. To ensure diverse and physically plausible interactions, we first uniformly sample contact points on the object surface and use the corresponding surface normal information provided by the simulator to parameterize action directions. Based on this procedure, we construct two primary non-prehensile action types: push, which applies outward forces to displace the object, and reach, which induces inward pulling interactions. This design enables systematic coverage of contact configurations while avoiding hand-crafted action heuristics.

c) Data preprocess: During data collection, we record the full mesh states of both objects and tools at each simulation time step, including rigid-body transformations and per-vertex deformations for soft objects. From each mesh, we uniformly sample surface points and apply farthest point sampling (FPS) to obtain fixed-size point clouds with well-distributed spatial coverage. These sampled point clouds provide consistent point correspondences across time and are used to construct object state sequences and tool action sequences as inputs to the world model.

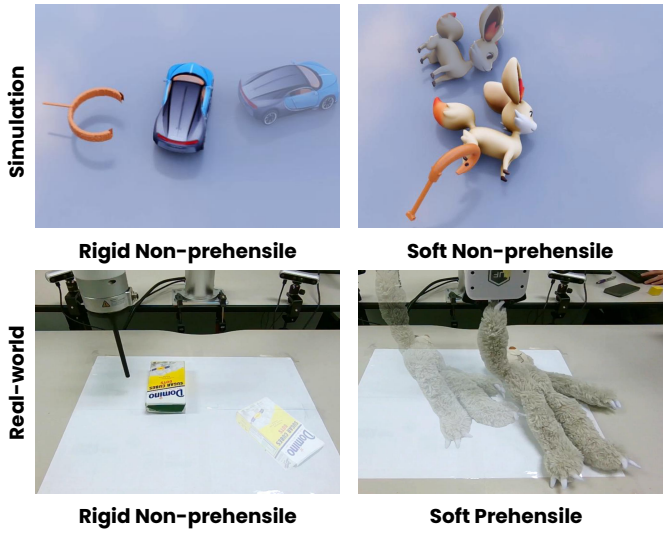


Fig. 1: **Benchmark tasks visualization**: Simulation and real-world benchmarks covering rigid and soft objects under non-prehensile and prehensile interactions, designed to evaluate generalization across diverse physical regimes.

TABLE I: **Gating Network Accuracy.** Expert selection accuracy (%) evaluated over 50 random interactions per task category. The gating network maintains consistently high accuracy across diverse regimes, indicating reliable expert routing for different physical dynamics.

	Push-X	Manipulate Toy	Soft Grasp	Overall
Acc.	95.4%	94.9%	99.8%	96.7%

d) Domain randomization: To improve generalization and reduce the sim-to-real gap, we apply extensive domain randomization during simulation. Specifically, we randomize physical properties such as mass and friction coefficients for rigid objects, as well as material parameters (e.g., Young's modulus) for deformable objects. In addition, to model actuation uncertainty commonly observed in real robotic systems, we inject Gaussian noise into each action at every time step.

e) Real-world data: To further improve generalization and bridge the sim-to-real gap, we additionally curate real-world interaction data from multiple sources, including [7, 6]. These datasets contain real object–robot interaction sequences captured with RGB-D sensors and reconstructed into 3D point

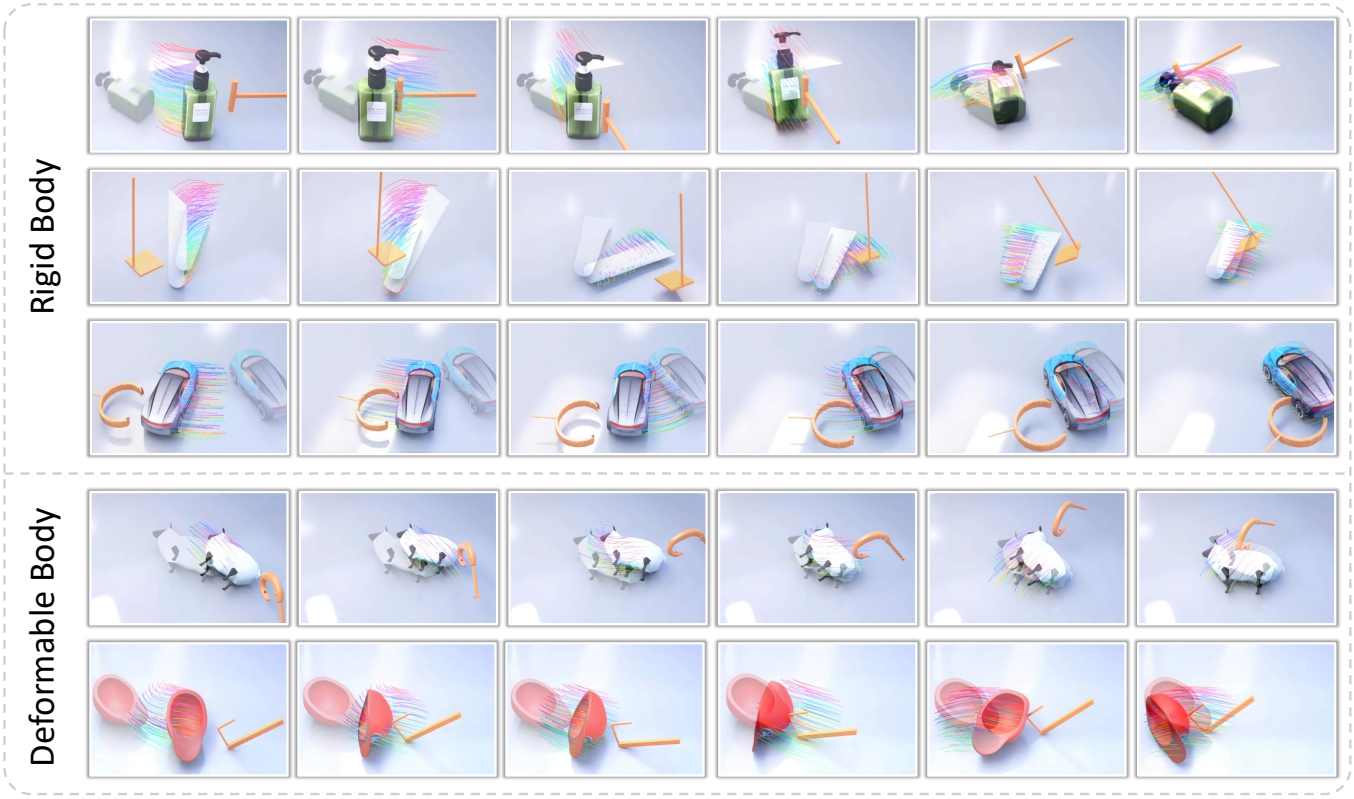


Fig. 2: **MPC rollout results in simulation tasks**. Qualitative simulation results of model-based control using world models trained with MoRE. The results demonstrates robust performance across diverse tools, objects, and physical regimes, including both rigid and deformable interactions.

clouds over time. Moreover, they include a wide range of prehensile manipulation behaviors that are difficult to faithfully model in simulation, particularly for deformable object grasping. We leverage these real-world demonstrations to train the prehensile experts in our model.

By unifying simulated and real data under a common 3D point-cloud representation and training with a balanced mixture, the model benefits from both the scale and controllability of simulation and the realism and noise characteristics of real-world observations.

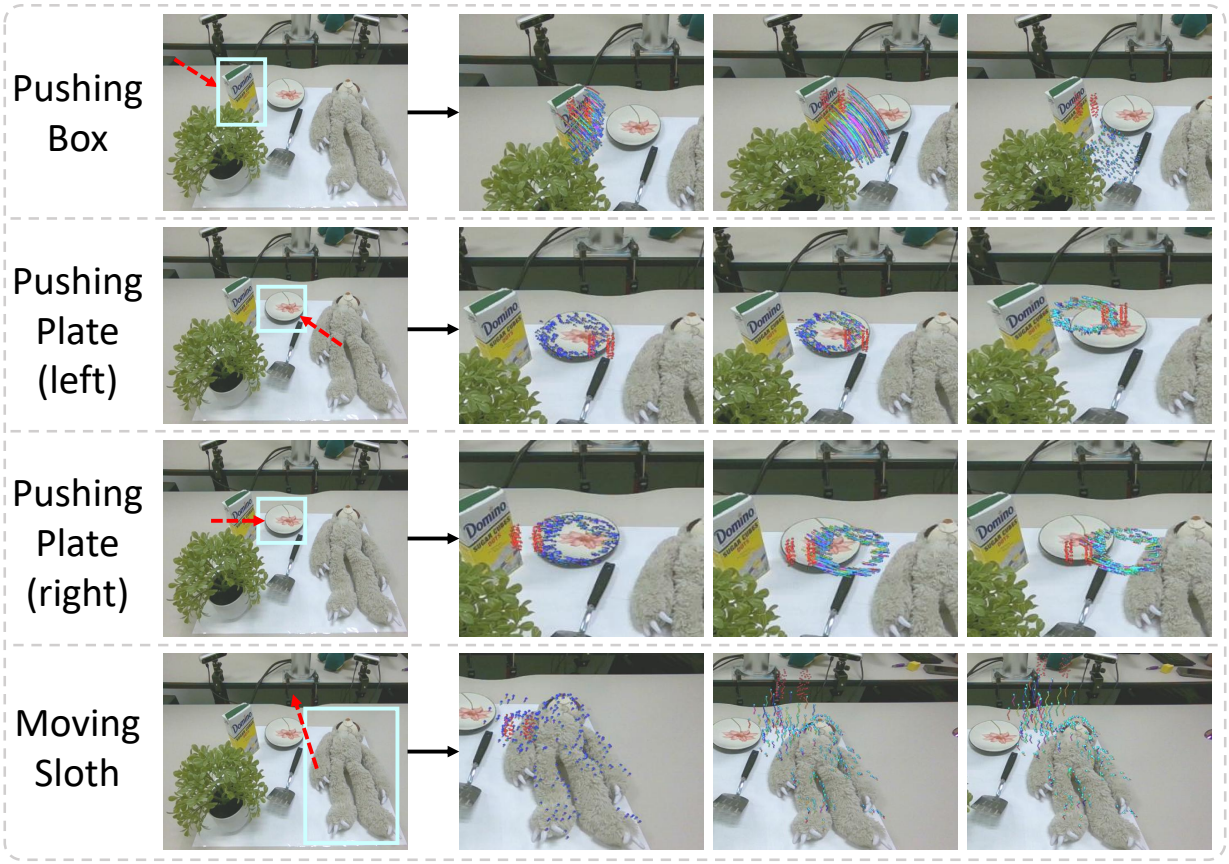
B. Model Architecture

a) Backbone: The model consists of two main components: a static geometry encoder and a motion encoder with a mixture-of-experts (MoE) design. The static encoder processes the current object point cloud using a frozen Uni3D [8] backbone, which encodes per-point geometric features. Uni3D is instantiated with grouped point tokens and a transformer-based architecture, producing 1024-dim features; its parameters are fixed during training to provide stable geometric priors. The motion encoder operates on dynamic tokens constructed from object and tool histories, and is implemented as a transformer-based network with multi-head self-attention operating on concatenated object and tool point tokens. The motion encoder outputs per-point 1024-dim motion features.

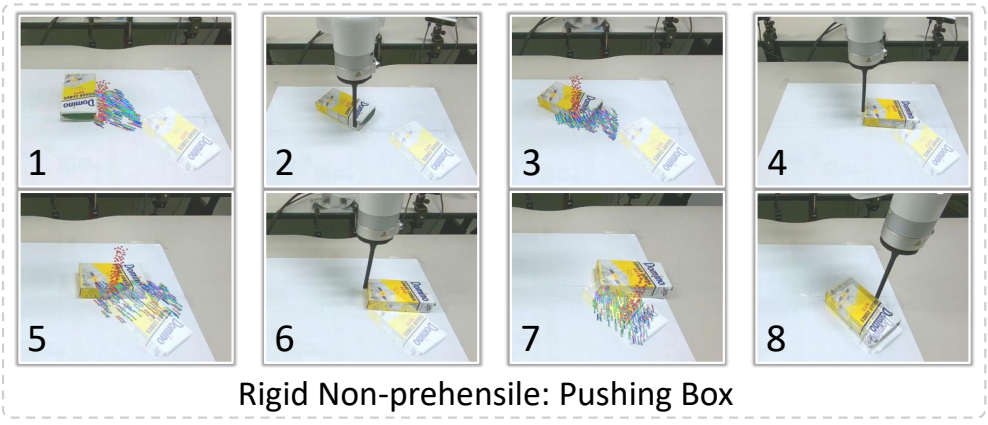
b) Fusion: We adopt a cross-attention mechanism to fuse static geometric features with motion features, enabling the resulting latent representations to encode dynamic motion information while preserving geometric consistency throughout the interaction process.

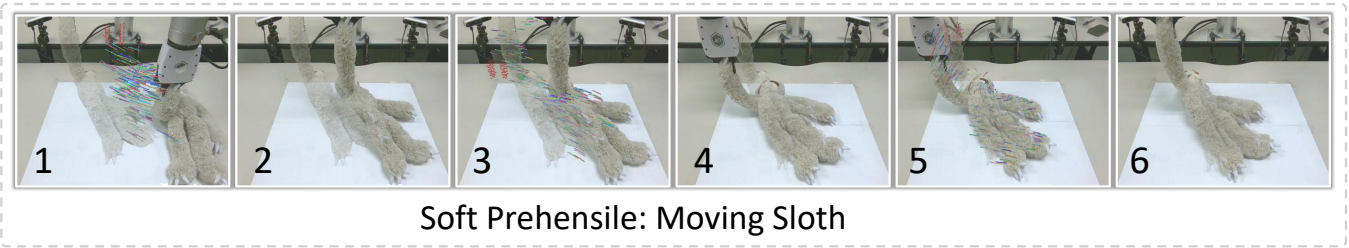
c) Decoding: To decode this latent representation into heterogeneous physical behaviors, we design K different motion experts. Each expert is an MLP head: the rigid expert predicts a global 6D rotation and 3D translation parameterizing an SE(3) transform, while the remaining experts predict per-point 3D displacements to model deformable dynamics. Given that different physical regimes exhibit substantially different dynamics patterns, we adopt hard expert routing. Specifically, we design the gating vector as a one-hot vector, which selects a single expert at each time step within an interaction sequence. The gating network is implemented as a two-layer MLP that takes the latent motion features as input and outputs expert selection logits. This design enforces explicit expert specialization and avoids blending incompatible dynamics across physical regimes.

d) Model forward process: Given a history of object and tool point clouds, the model performs a forward prediction as follows. The current object point cloud is first encoded by the static geometry encoder to obtain per-point geometric features. Object and tool histories are flattened along the temporal



(a) Real-world dynamics prediction under heterogeneous physical properties, demonstrating that our unified model can accurately capture object dynamics across different material regimes.





(b) MPC rollouts on real-world manipulation tasks, showing robust control performance under sensor noise and partial observations, and highlighting the model's ability to handle contact-rich interactions across diverse physical regimes.

Fig. 3: **Real-world results on diverse manipulation tasks.**

dimension and embedded into dynamic tokens, which are processed by the motion encoder to produce motion features. These static and motion features are fused through cross-attention, producing geometry-aware motion representations. The gating network predicts expert weights, and each expert produces a candidate motion prediction. The rigid expert applies a global SE(3) transform to all object points, while deformable experts predict per-point displacements. The final output is selected based on the routing weights from the gating module.

C. MPC Inference Pipeline

Since expert routing in MoRE depends on historical object dynamics, the world model cannot directly infer the appropriate physical regime from static observations. Therefore, we adopt a two-stage MPC procedure.

a) Initialization: We first apply a short sequence of random interactions to the object to get observable motion responses. These probing interactions provide the dynamic history required by the gating network, which then predicts the expert corresponding to the underlying physical regime.

b) MPC procedure: Conditioning on the selected expert, we perform MPC rollouts using the learned world model. For non-prehensile manipulation, we sample 48 candidate push actions by selecting 16 contact points on the object surface and 3 pushing directions per point. For prehensile manipulation, after a grasp is established, we uniformly sample candidate tool displacements within a bounded workspace. Each candidate is represented as a short "start–end" waypoint trajectory for the tool. For each candidate, we convert the start–end tool motion into a dense action sequence by interpolating the end-effector pose over a fixed horizon. Given the current history buffer of object states and actions, we roll out the world model autoregressively for a fixed number of steps. At each step, the predicted object point cloud is appended to the history by shifting the buffer, while the corresponding future action is updated from the precomputed action sequence. After generating trajectories for all candidates, we compute the goal cost (Chamfer distance) and select the best candidate and the rollout step number (it is often better to stop early rather than complete the full action sequence). The corresponding action is executed and the system replans at the next observation.

c) Real-world execution: In real-world experiments we normalize the coordinates by the mean of the object points so that the point clouds match the model's training distribution. The object point cloud is obtained by fusing point clouds from four Realsense D455 cameras, each masked using SAM3 [1] with text prompts ("box", "sloth"). Statistical outlier removal from Open3D [9] is used to remove points based on average neighbor distance. Farthest point sampling is applied in the same way as during training (512 object points, 100 end-effector points).

D. Gating Network Evaluation

In this section, we provide a detailed evaluation of the gating network. The gating network determines which expert is selected and therefore plays a critical role in the accuracy of the final predictions. We evaluate its performance on three categories of tasks: rigid non-prehensile, soft non-prehensile, and soft prehensile manipulation. For each category of tasks, we perform 50 random interaction trials and measure the expert selection accuracy. As shown in Table I, the gating network can effectively infer the appropriate expert from historical object dynamics, enabling accurate expert routing and high-fidelity dynamics prediction.

II. Additional Qualitative Results

A. Benchmark Task Visualization

We visualize representative samples from the four benchmark tasks used in our evaluation in Fig. 1. The tasks span rigid and soft objects under non-prehensile and prehensile interactions across both simulation and real-world settings, covering diverse contact dynamics and interactions.

B. Simulation Visualization

In this section, we provide additional simulation results on model-based control using world models trained with MoRE. We evaluate our approach on different non-prehensile manipulation tasks involving diverse rigid objects, deformable objects, and various tools. The qualitative results, shown in Figure 2, demonstrate that MoRE enables accurate dynamics prediction, robust handling of heterogeneous physical properties, and generalization across diverse object–tool interactions.

C. Real-world Visualizations

We visualize real-world model-based control results in Fig. 3. Our results show strong model performance under sensor noise and partial observations.

III. Limitations

Despite strong performance across a range of interaction regimes, our method has several limitations related to expert compositionality, geometric representation, and physical expressiveness.

a) Single motion expert activation.: Our model assumes that a single motion expert is active at each time step. While this design is sufficient for the single-object, single-material interactions studied in this work, it may limit direct applicability to more complex scenarios involving multiple interacting objects or mixed material properties. However, this assumption is not inherent to the model architecture. In multi-object settings, expert routing can be performed independently for each object, enabling parallel prediction of object-specific dynamics and naturally extending our framework to multi-object manipulation.

b) Deterministic prediction.: Our approach captures coarse physical regimes (e.g., rigid versus deformable) but does not explicitly model finer-grained physical properties such as mass, friction, density, or elasticity. These unmodeled factors introduce inherent variability in real-world contact

dynamics, yet our model predicts interactions deterministically, producing a single outcome per interaction. Incorporating richer physical attributes and generative dynamics is a promising avenue for future research.

REFERENCES

[1] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[2] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025. URL https://arxiv.org/abs/2506.18088.

[3] Chunru Lin, Haotian Yuan, Yian Wang, Xiaowen Qiu, Tsun-Hsuan Wang, Minghao Guo, Bohan Wang, Yashraj Narang, Dieter Fox, and Chuang Gan. Robotsmith: Generative robotic tool design for acquisition of complex manipulation skills. *arXiv preprint arXiv:2506.14763*, 2025.

[4] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac Lab - A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning. *arXiv preprint arXiv:2511.04831*, 2025. doi: 10.48550/arXiv.2511.04831. URL https://arxiv.org/abs/2511.04831.

[5] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. 2021.

[6] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. *arXiv preprint arXiv:2407.07889*, 2024.

[7] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. *arXiv preprint arXiv:2506.15680*, 2025.

[8] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[9] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.