JODA: Composable Joint Dynamics for Articulated Objects

Tianhong Gao* Peking University
Cheng Yu* Peking University
Yinghao Xu Ant Group, Robbyant
Mengyu Chu† Peking University

Abstract

Articulated objects used in simulation and embodied AI are typically specified by geometry and kinematic structure, but lack the fine-grained dynamical effects that govern realistic mechanical behavior, such as frictional holding, detents, soft closing, and snap latching. Existing approaches either ignore the detailed structure of dynamics entirely or use simple models with limited expressiveness.

We introduce JODA, a framework for generating joint-level dynamics as a structured three-channel field over the joint degree of freedom, capturing conservative forces, dry friction, and damping. Instantiated using shape-constrained piecewise cubic interpolation (PCHIP), this formulation defines a compact, expressive, and interpretable function space that supports both direct editing and optimization-based refinement. Building on this representation, we develop an agentic generation pipeline for inferring and refining joint dynamics from multimodal inputs. Given visual observations and joint context, the pipeline uses a vision-language model to propose structured dynamical primitives, composes them into the three-channel dynamics field, and iteratively revises the proposals through diagnostic feedback. We demonstrate that JODA enables plausible and controllable modeling of diverse joint behaviors, providing a unified interface for inference, editing, and optimization. Code and examples are available on the project page: `https://guesss2022.github.io/JODA/`.

1 Introduction

Articulated-body assets are a central substrate for physical simulation, robotics, embodied AI, and interactive scene generation. Recent asset pipelines [Mandi et al., 2025, He et al., 2026] can recover or synthesize increasingly rich visual geometry and kinematic structure: parts are separated, joints are assigned axes and ranges, and assets can often be loaded by a physics engine. However, real-world interaction of articulated objects depends on more than kinematics alone. A laptop lid holds at many intermediate angles; a microwave door may pop through a closing latch; a refrigerator door can be pulled into its final closed pose; an oven door may return under a broad spring-like tendency while being resisted by friction. These behaviors reflect not only how objects are geometrically connected, but more importantly, how they behave dynamically. Such dynamic effects are typically absent from automatic pipelines and must instead be manually authored or carefully tuned at significant cost, making them difficult to scale across datasets, scenes, and applications. As a result, automatic pipelines remain largely restricted to geometry and kinematic structure, leaving physical dynamics poorly modeled and leading to a persistent sim-to-real gap.

This paper studies the problem of generating *dynamical structures* for articulated-body assets. By dynamical structures, we mean a reduced-order representation of joint-level mechanical effects that complements geometry and kinematics, without attempting to model the complete dynamics of the

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

object and its environment. Unlike a prescribed motion trajectory, this representation specifies how internal joint effects vary with the joint state and thereby influence motion under different initial conditions and applied forces or torques. Other factors, including inertia, gravity, joint limits, contacts, and environmental interactions, are handled separately by the asset model and surrounding simulator. Rather than recovering the exact hidden mechanical assemblies inside an object (e.g., specific springs or gears), we model their equivalent effects at the joint level. For example, mechanisms such as magnetic catches or spring latches may not be explicitly encoded by the asset geometry, yet their behavior can be expressed as position-dependent forces, friction, or damping over the joint degree of freedom. This unified representation abstracts away low-level mechanical complexity while preserving macroscopic interactive fidelity.

A central question is how such structured dynamics can be initialized without direct supervision or manual design. Interestingly, we find that vision-language models (VLMs) capture useful implicit physical priors about object behavior; for instance, a VLM can infer an oven's tendency toward a stable closed configuration. However, these priors are latent and unstructured. Our key insight is that the challenge is not in eliciting physical intuition from VLMs, but in grounding it within a structured, composable representation suitable for simulation.

We introduce JODA, a framework that combines a composable dynamics representation with an agentic generation pipeline for inferring and refining joint dynamics from multimodal inputs. Given an articulated asset and visual observations across joint configurations, a VLM proposes joint-level dynamical effects as structured hypotheses. These hypotheses are composed into a unified three-channel dynamics field, and a diagnostic visualization of the resulting field is fed back to the VLM for iterative proposal revision. The resulting representation is compact, interpretable, and compatible with physical simulation, and supports both semantic and continuous refinement, including direct editing and gradient-based optimization. Our approach unifies representation, inference, and optimization within a single framework:

- **Structured Joint Representation.** A composable representation for joint dynamics as a structured function space over the joint degree of freedom, enabling expressive yet interpretable modeling of forces, friction, and damping.
- **Agentic Generation.** An agentic pipeline that leverages VLMs to instantiate and iteratively refine joint dynamics, enabling reliable and diverse modeling.
- **Unified Interaction and Optimization.** JODA supports flexible interaction and refinement, including direct or semantic editing and gradient-based optimization via differentiable simulation, providing a unified interface for inference, editing, and optimization.

2 Related work

Articulated object and asset generation Articulated object research has progressed from mobility prediction and part-level datasets to automatic URDF-like reconstruction and generation from images, videos, or language [Hu et al., 2017, Xiang et al., 2020, Chen et al., 2024, Liu et al., 2025, Mandi et al., 2025, Le et al., 2025, Xia et al., 2025, He et al., 2026]. These methods primarily recover the geometric and kinematic scaffold of an asset: parts, hierarchy, joint type, axis, origin, and motion limits. JODA is complementary: it assumes this scaffold is available and generates the missing joint-level dynamical structure that determines how the articulation behaves during simulation.

Simulation-ready assets and differentiable physics Physics simulators provide the substrate for executable articulated objects [Xiang et al., 2020, Todorov et al., 2012, Makoviychuk et al., 2021, MuJoCo Team, 2024, Freeman et al., 2021, Howell et al., 2022]. They expose primitives such as PD-drive controllers, tendons, and runtime user-specified generalized forces. The built-in mechanisms are well suited to motors or tendon actuation, but are not a convenient way to express everyday object dynamics such as detents, endpoint attraction, or soft-close damping; using external forces still requires hand-designed force laws. Recent work moves beyond these defaults: ArtVIP realizes realistic behaviors in professionally authored assets through position-dependent extensions of Isaac Sim joint drives and embedded modular interaction behaviors [Jin et al., 2026]. While physically meaningful, this drive-centric formulation exposes the resulting joint effects only indirectly through coupled stiffness, targets, damping, friction, and state-dependent logic, making effect-level inspection, editing, and generation less direct.

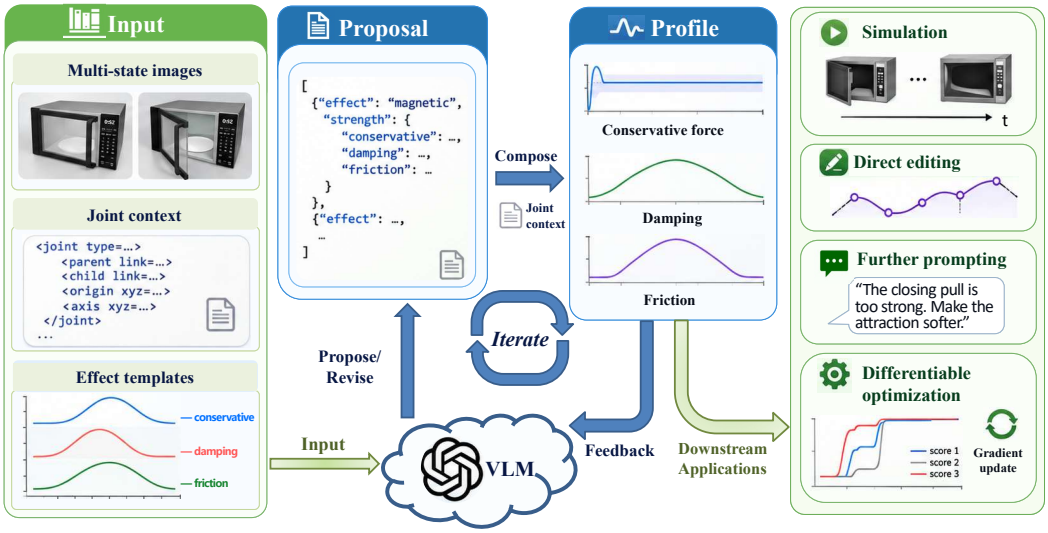


Figure 1: Overview of JODA. A vision-language model proposes and iteratively refines structured joint effects from multimodal inputs, which are composed into a three-channel dynamics profile enabling simulation, editing, and optimization.

Asset-centric benchmarks apply such simulators to large collections of interactive objects and tasks [Gu et al., 2023, Li et al., 2024]. Simulation-trained robot policies instantiate the same pattern [Zakka et al., 2023, Zhang et al., 2024]. Their articulated assets typically rely on simple target-position control, equivalent to adding springs, together with global friction and damping parameters; this is executable but has limited expressiveness for structured joint dynamics. JODA uses the simulator as the execution substrate, but directly exposes joint effects through an interpretable and expressive field representation that can be generated, edited, simulated, and optimized.

Joint-level interaction effects Measurements of everyday doors and drawers show diverse force trajectories with low-dimensional structure, suggesting the potential for compact approximations [Jain et al., 2010]. More recently, Hoi! provides multimodal observations and force measurements of articulated-object interactions, but leaves open how these measurements can be distilled into a compact, executable representation of joint dynamics [Engelbracht et al., 2026]. In engineering mechanics, reduced or equivalent force models are also a central tool for representing complex joint interfaces without simulating every contact detail [Brake, 2018]. Thus, rather than reconstructing internal mechanisms, JODA models their equivalent joint-level effects. Industrial hinge and lid-support catalogs organize such effects into recurring families, including detent, bistable, torque, counterbalance, soft-down, and lift-assist mechanisms [Southco, 2021, Sugatsune, n.d.]. They provide engineering grounding for our templates, which compose behavioral priors into force, friction, and damping fields rather than identifying the exact hidden mechanism.

3 Method

Given joint context and visual observations, JODA uses an agentic generation pipeline to infer a structured dynamical representation that captures realistic interaction behavior. The inferred representation can subsequently be refined through interactive editing or simulation-based optimization. Fig. 1 gives an overview of our pipeline.

3.1 Problem Formulation

The input to JODA is an articulated asset with a selected target joint. The asset provides a visual model, a kinematic hierarchy, and a finite joint range. From this asset, we extract joint context, including joint type, parent and child bodies, joint axis, joint range, equivalent inertia or mass, and gravitational effects along the degree of freedom (DOF), which can be directly obtained from standard simulation or modeling tools like MuJoCo.

The output represents the internal dynamical structure of that joint. It is internal in the sense that it represents object-specific effects such as hinge drag, latching, or return tendencies, not external forces such as gravity. These effects are expressed as generalized forces or coefficients along a single DOF. Joint-limit behavior is modeled separately, as it is conventionally defined at the kinematic level as boundary conditions.

3.2 Joint-Level Dynamical Representation

Let q denote the generalized coordinate of the target joint, with finite range $[q_{\min}, q_{\max}]$. We use a normalized coordinate

$$s = \frac{q - q_{\min}}{q_{\max} - q_{\min}}, \quad s \in [0, 1].$$

JODA represents the internal dynamical structure of a joint with three position-dependent fields:

$$F_{\text{cons}}(s), \quad F_{\text{fric,max}}(s), \quad C_{\text{damp}}(s).$$

The conservative channel F_{cons} models position-dependent generalized forces or torques, such as spring return, magnetic-like attraction, detent wells, snap-through barriers, and bistable tendencies. The dry-friction channel $F_{\text{fric,max}}$ gives the local friction bound. At rest, the friction force can take any value within $\pm F_{\text{fric,max}}(s)$ to resist incipient motion; during sliding, it opposes the velocity with magnitude $F_{\text{fric,max}}(s)$. We approximate the maximum static and kinetic friction magnitudes as equal. The damping channel C_{damp} models a velocity-dependent resistive coefficient, producing a damping force proportional to $-C_{\text{damp}}(s)\dot{q}$.

Each channel is composed from local effect components. A component j is defined by an active interval $[a_j, b_j] \subseteq [0, 1]$, a set of local control-point positions $x_{j,k} \in [0, 1]$, and values $y_{j,k}$. For $s \in [a_j, b_j]$, the local coordinate is

$$u = \frac{s - a_j}{b_j - a_j}.$$

The component value is obtained via shape-constrained piecewise cubic interpolation (PCHIP; Fritsch and Butland, 1984), producing smooth curves over local control points while preserving physically plausible profiles. Outside $[a_j, b_j]$, the component contributes zero. The channel-level field is obtained by summing or taking a maximum over active components. This representation is low-dimensional, local, interpretable, and editable: changing a few control-point values or an active interval changes a specific portion of the joint behavior without requiring a global curve redesign.

3.3 Effect Template Library

JODA uses a composable library of *effect-templates*. The term effect is deliberate: a template is not a claim that a particular hidden mechanical mechanism exists inside the object. Instead, it specifies an equivalent joint-level behavior that may be produced by many physical designs. For example, an endpoint attraction template may approximate a magnetic catch or a spring-loaded return mechanism. This mechanism-agnostic design makes our effect vocabulary more transferable.

Each template contains semantic metadata and prototype curves for the three channels. The metadata includes the effect name, a short description, and typical placement priors. The prototype curves define the shapes of the conservative-force, friction, and damping contributions in local normalized coordinates. Instantiating a template means placing it on a selected interval of the target joint and scaling each channel.

The current library covers a range of common joint behaviors, including constant friction, constant damping, constant directional conservative force, endpoint magnetic-like return, spring-like return, snap detent, internal detent, and bistable effects. A summary of representative templates and their corresponding behaviors is provided in App. C. Higher-level interaction behaviors can be decomposed into these primitive effect templates. For example, a free-stop experience is typically represented by strong constant friction, possibly accompanied by damping.

3.4 VLM-guided Effect Proposal

The VLM stage predicts structured effect proposals. Its inputs are the joint context, rendered images at multiple joint states, and effect-template previews. The prompt asks the VLM to reason about the

felt behavior a user would experience while moving the joint across its range. The VLM is explicitly instructed not to output final control points or force curves. Instead, it outputs JSON containing effect proposals. A summary of the prompt design is provided in App. C, with the full prompt and schema in the supplementary material.

Each effect proposal specifies an effect name from the library, an active interval (`start_ratio`, `end_ratio`), and qualitative strength labels for the conservative, friction, and damping channels. The proposal also includes descriptions like reasons and evidence which are useful for inspection.

The output separates semantic choice from numerical realization. This division reduces the burden on the VLM: it need not synthesize physically scaled curves directly. It only chooses which equivalent effects should exist, where they act, and how strong they are qualitatively.

3.5 Compilation to Dynamics

The compiler maps VLM proposals and joint context into a concrete multi-component field. It converts normalized intervals to real joint intervals and scales each channel into physical units.

The compiler is semi-deterministic. Qualitative strength labels are mapped to numeric multipliers by sampling within predefined bands, while the sample is generated from a stable key containing some meta-info. This preserves repeatability while avoiding one-to-one mapping from strength labels to a deterministic value.

To set the physical scale, the compiler computes reference magnitudes from joint context. For conservative force and friction, it uses the larger of an estimated gravity scale and an inertial scale based on equivalent inertia or mass, joint range, and a reference travel time. The damping reference is chosen so that multiplying it by a reference speed yields the reference force. Each template channel is then scaled so that its prototype peak reaches the target channel reference multiplied by the sampled multiplier. The result is a `composed.json` file containing the final three-channel PCHIP field model. This file is the source for downstream plotting and simulation.

3.6 Agentic Generation with Diagnostic Feedback

After a proposal is compiled, JODA renders a diagnostic visualization of the resulting joint-level dynamics and feeds it back to the VLM. This visualization (e.g., Fig. 2c) illustrates the composed effects across the joint DOF and provides a structured signal for assessing whether the generated dynamics match the intended behavior. We provide detailed interpretations of the diagnostic visualization in Sec. 4.2 alongside concrete examples. Based on this feedback, the VLM can iteratively revise the proposal by adjusting the effect types, active intervals, strength labels, or channel-specific refinement factors. This closed-loop iteration enables the system to progressively align the generated dynamics with the desired interaction behavior.

3.7 Interactive and Differentiable Refinement

Refinement after VLM generation may operate on variables including control-point values, channel strengths, active-interval anchors, and joint-limit damping parameters. JODA supports three refinement modes.

Direct editing. An experienced user can directly modify the values mentioned above.

Human-supervised VLM refinement. A user can describe a problem in language, such as “the final closing pull is too strong” or “the middle range should hold better”. The VLM then revises its proposals while preserving the overall intended motion when appropriate. The user can also specify the desired effects over the full DOF, guiding the model to choose among multiple plausible effects.

Differentiable simulation optimization. VLMs provide strong semantic priors for motion, which can be further refined when higher physical accuracy is required. While manual adjustment is possible, it can be time-consuming, especially under quantitative constraints. Differentiable simulation offers a general mechanism for optimizing joint profiles under diverse supervision signals. In particular, leveraging real-world interaction trajectories as supervision targets enables direct alignment with observed motion, eliminating the need for manual tuning.

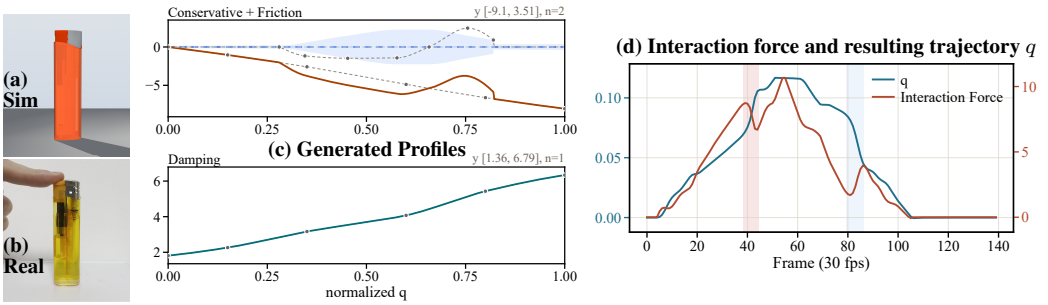


Figure 2: **Lighter-button dynamics:** (a) simulated and (b) real-world visuals; (c) JODA-generated force and friction profiles (see Sec. 4.2 for plot interpretation); (d) Force and trajectory of the virtual interaction.

4 Experiments

4.1 Experimental setup and baselines

We evaluate JODA on nontrivial object categories in PartNet-Mobility [Xiang et al., 2020] and Lightwheel Sim-Ready Assets [Lightwheel, 2025], processing one target joint per run. We use two API models, `gemini-3.1-pro-preview` with the default reasoning setting and `gpt-5.4` with high reasoning; we did not observe a clear difference in task-level capability between them. Although the representation is simulator-agnostic and has also been verified in Isaac Sim, all experiments reported here use the MuJoCo/MJX backend [Todorov et al., 2012, MuJoCo Team, 2024]. All experiments are run on a laptop with an AMD Ryzen 9 8945HX CPU (16 cores/32 threads) and 32 GB of RAM. Implementation details are provided in App. A.

Our first baseline removes the generated internal dynamical structure and simulates the same articulated asset only with simple constant friction or damping. The second kind of baseline adds a linear return force over the relevant joint interval, i.e., a spring-style model commonly used in simulation-trained manipulation setups [Zakka et al., 2023, Zhang et al., 2024].

4.2 Case analysis

We visualize the generated dynamics using a diagnostic profile (e.g., Fig. 2c). This profile summarizes the composed joint-level dynamics across the DOF and is also used by the VLM to determine whether further refinement is needed. In this visualization, the red curve represents the conservative force, while the blue dashed line indicates the gravity-induced equilibrium. The shaded band reflects the friction range, within which small force imbalances may not lead to motion. The gray dashed curves show the individual conservative-force contribution of each instantiated template before composition, and the dots mark its PCHIP control points over the active interval. When the red curve lies above the blue dashed line, the net effect drives the joint in the positive direction; when it lies below, it drives motion in the negative direction. This visualization provides an interpretable link between generated dynamics and the resulting behavior. The supplementary video shows the cases below and additional behaviors across all 21 tested PartNet-Mobility object types.

Lighter button We first examine the push button of a cigarette lighter, whose target joint is a short-travel prismatic DOF, as shown in Fig. 2a. Fig. 2c shows the compiled profile. The dominant effect is a full-range spring-like return toward the unpressed state, while an additional internal snap-like structure creates a non-monotone modification of the restoring force in the middle part of the travel.

To probe the resulting interaction behavior, we use a virtual-hand interaction. This interaction mode is designed to mimic a human hand: it moves a bounded, rate-limited control target, and the simulator generates the corresponding interaction force needed to push or release the button to reach that target using PD control. Fig. 2d plots the resulting joint trajectory and interaction force over time.

The characteristic behavior of this button is that the internal mechanism causes the effective rebound force to vary non-monotonically over the whole DOF. As a result, both pressing and releasing can produce short intervals of sudden high-speed motion, which is consistent with the familiar push-

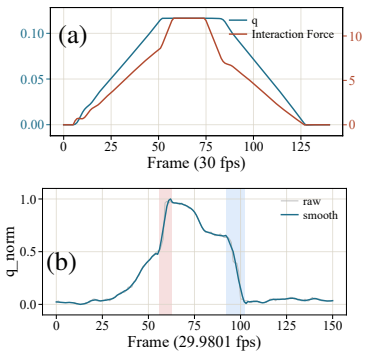


Figure 3: Lighter button trajectories: (a) spring baseline; (b) real-world data (`q_norm`: raw, `smooth`: smoothed).

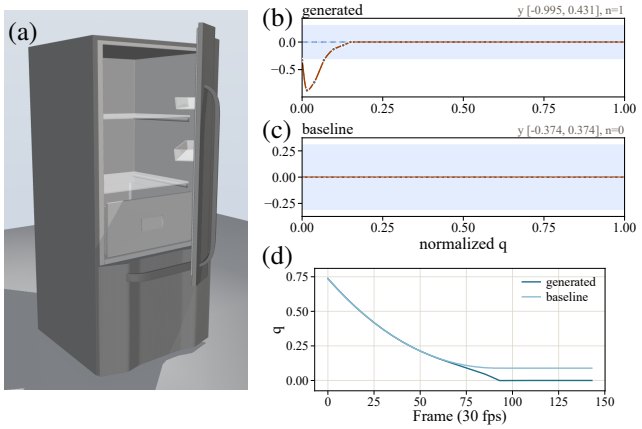


Figure 4: Refrigerator door: (a) visuals; (b, c) profiles; (d) trajectories.

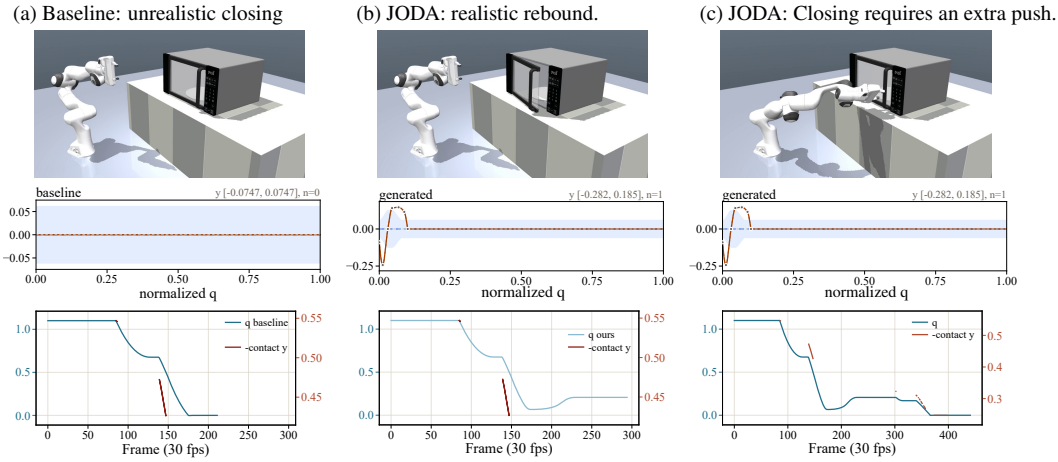


Figure 5: Microwave door dynamics (visuals, profiles, and trajectories). (a) Constant-damping baseline: the door automatically closes under slight pushes, which is unrealistic. (b) JODA with the same robot motion offers a rebound near closure, preventing latching. (c) JODA with an additional push: the door successfully latches after an extra contact.

through and pop-back feel of a lighter button. The shaded regions in Fig. 2d mark these sudden high-speed motion intervals.

We compare this result with a simple-spring baseline that uses only a broad return spring, without the generated snap-like internal structure. The baseline is evaluated with the same virtual-hand interaction controller. As shown in Fig. 3a, the simple spring restores the button but misses the distinctive push-through and pop-back transients.

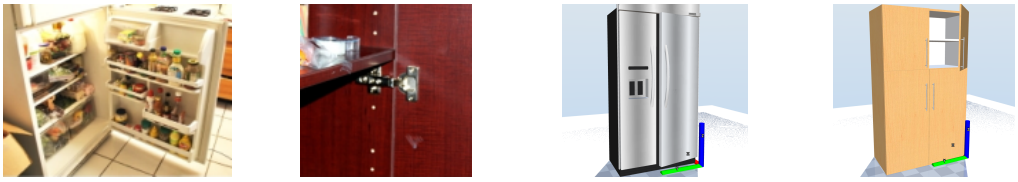
We also use a real lighter, shown in Fig. 2b, to verify that the button motion under human pressing exhibits a similar push-through and pop-back pattern. As shown in Fig. 3b, the simulated trajectory and the real trajectory share the same qualitative behavior. The shaded regions indicate the two corresponding sudden high-speed motion intervals.

Refrigerator door We evaluate a refrigerator door (Fig. 4a). Our model captures a magnetic-seal effect that assists closing near the shut configuration, enabling the door to automatically close when nearly closed. In contrast, the baseline lacks such a structure and fails to close under the same conditions. Fig. 4d compares the resulting trajectories, while Fig. 4b,c show the generated and baseline profiles, respectively.

Microwave door We evaluate the generated dynamics on a microwave door (Fig. 5). To evaluate interaction, we simulate a robot manipulator executing predefined pushing trajectories. JODA generates a profile that captures a spring-loaded latch near the closed configuration, introducing an energy barrier that must be overcome to fully close. We compare JODA with a constant drag baseline. Under the same slight push, the baseline exhibits unrealistic door closing in a simple slowing-down motion (Fig. 5a), while our model produces a partial door closure with a rebound behavior near the end (Fig. 5b). Our model requires an additional push to fully latch (Fig. 5c), which is consistent with real-world interactions. These results highlight the importance of physically faithful dynamics for reducing sim-to-real discrepancies and enabling better real-world transfer.

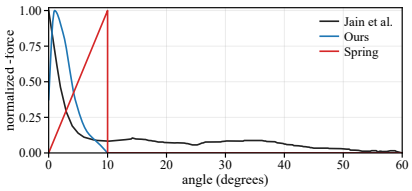
4.3 Quantitative results

Our representation defines a joint-level dynamics profile, from which we can derive the quasi-static force that opens the door at each configuration. While this profile represents intrinsic joint behavior and is not directly measurable, prior work by Jain et al. [2010] measured the corresponding quasi-static opening forces for everyday objects (e.g., refrigerator doors) in real-world settings. We compare the forces derived from our generated profiles with these measured curves. Specifically, we normalize each curve to $[0, 1]$ and evaluate JODA against a linear spring baseline on a common 0–60° grid. Fig. 6 shows the objects used for this comparison, and the first two images are real object photos from Jain et al. [2010]. Fig. 7 compares the quasi-static opening forces (positive values indicate external forces that increase the angle), and Fig. 7c reports the RMSE. JODA more closely matches the measured curves, whereas a linear spring fails to capture the structured, nonlinear behavior, as it is constrained to produce zero force at equilibrium and linear force variation with displacement. This demonstrates that expressive joint-level dynamics are necessary to reproduce real-world forces across configurations beyond simple models.

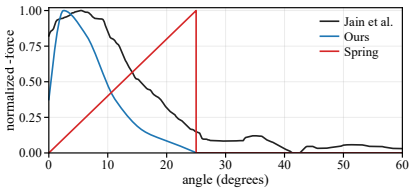


(a) Jain fridge. (b) Jain cabinet. (c) Our fridge render. (d) Our cabinet render.

Figure 6: Visual references for quantitative comparison.



(a) Refrigerator door.



(b) Kitchen cabinet door.

	Fridge	Cabinet
Spring	0.240	0.386
JODA	**0.121**	**0.180**

(c) Normalized force error w.r.t. Jain et al. [2010]. (Lower is better.)

Figure 7: Quasi-static opening force comparison against measurements from Jain et al. [2010].

We also measure computational cost. With a timestep of $dt = 0.005$s, across all runs with valid timing records, JODA takes 0.062 ms per simulation step on average, which is 1.37x the original MuJoCo cost, indicating only a modest simulation overhead. For VLM proposal, across 252 proposal results, each iteration uses on average 25.5K input tokens and produces 4.7K output tokens, where output tokens include reasoning tokens.

4.4 Experiments with measured interactions

Differentiable refinement: a case study. We experiment with a cabinet door that opens under gravity (Fig. 8b). In Fig. 8e, the model-generated result provides too much support near the high-q end, preventing the door from opening naturally. We sample three trajectories from a real cabinet instance (Fig. 8a; Fig. 8c, natural release at frame 0). Starting from the generated profile, we optimize its conservative-force control-point values through differentiable MJX rollouts by minimizing the average joint-position MSE across the three measured trajectories. After 9 Adam iterations, the loss

decreases from 4.10×10^{-2} to 1.56×10^{-3}. The resulting optimized profile allows the simulated door to open naturally while retaining part of the support effect, yielding a more reasonable cabinet profile (Fig. 8e). The original and optimized profiles are rolled out from a held-out initial condition, and the resulting simulated trajectories are compared in Fig. 8d.

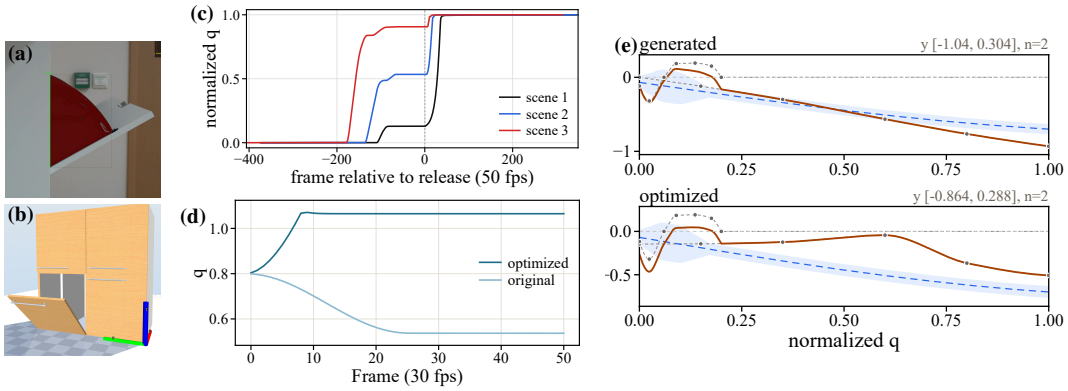


Figure 8: Differentiable refinement. (a,b) real and simulated visuals; (c) target trajectories; (d) release at q=0.8: original closes, optimized matches the opening behavior; (e) profiles.

Dataset-scale refinement and evaluation. To complement the single-object case study with a broader quantitative assessment, we evaluate closing-motion prediction and inverse effort on 48 articulated objects from the Hoi! dataset [Engelbracht et al., 2026], fitting each object's model using its single opening interaction. Across the 48 objects, zero-shot JODA achieves a lower object-macro mean normalized trajectory RMSE (NRMSE; 0.2191) than classical identification (0.4051) and the adapted residual GP (0.4219). Refinement reduces the mean to 0.2073 (5.4%) and inverse-effort NRMSE from 0.7070 to 0.6737 (4.7%; Table 1). Trajectory error decreases on 30 objects and increases on 18. Dataset processing, simulation setup, refinement and baseline details, metric definitions, and additional results are provided in App. B.

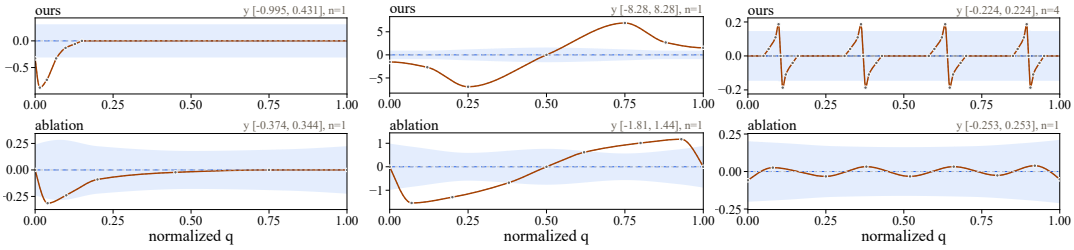
Table 1: Closing-motion prediction and inverse effort on 48 objects from the Hoi! dataset. Lower object-macro NRMSE is better.

Method	Revolute (25)	Prismatic (23)	All (48)
Trajectory NRMSE			
Classical identification	0.5587	0.2381	0.4051
Residual GP	0.2864	0.5691	0.4219
JODA zero-shot	0.2666	0.1674	0.2191
JODA optimized	**0.2511**	**0.1596**	**0.2073**
Inverse-effort NRMSE			
Classical identification	1.1369	0.9382	1.0417
Residual GP	1.6987	5.4167	3.4802
JODA zero-shot	0.5845	0.8401	0.7070
JODA optimized	**0.5778**	**0.7779**	**0.6737**

Together, these experiments support the use of JODA's generated fields as a prior and an expressive representation of joint dynamics. The differentiable refinement capability introduced in Sec. 3.7 then supports both direct multi-trajectory fitting and regularized adaptation from sparse measured interactions.

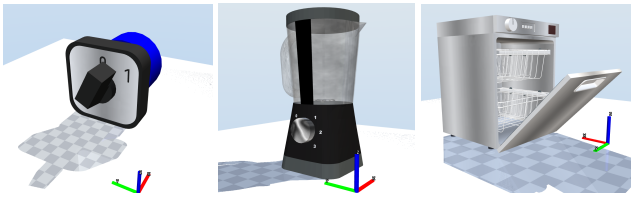
4.5 Ablation studies

We first ablate the template library by directly asking the VLM to output numerical control-point coordinates and then interpolating those points into fields. This setting places a much heavier burden on the VLM's quantitative reasoning and curve-generation ability: it often fails to choose appropriate magnitudes, and without engineering priors it also introduces small but behaviorally important defects.



(a) Refrigerator door. (b) Switch. (c) Selector knob.

Figure 9: Our results vs. No-template ablation results.



(a) Switch (b) Selector (c) Dishwasher

Figure 10: Prompt images used in the ablation.

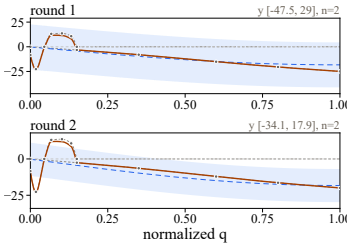


Figure 11: Initial and iterated field profiles for a dishwasher door.

In the refrigerator example (Fig. 9a), the magnetic-attraction force is too weak and acts over an overly broad range. In the switch example (Fig. 9b; Fig. 10a), the conservative force is weak relative to friction, and the force dropping to zero at the balance point makes the return tendency too weak. In the selector-knob example (Fig. 9c; Fig. 10b), the conservative force is again too weak relative to friction. In all these examples, both JODA and the ablations are evaluated without iterative rounds. Furthermore, removing the template library significantly degrades reproducibility.

The VLM iteration step is especially useful in cases in which gravity affects joint motion. In the dishwasher-door example (Fig. 10c), Fig. 11 shows that the round-1 near-closed attraction does not overcome friction, while the opening-end assist is too large and creates an early closing tendency that may prevent the door from fully opening. After iteration, the far-end assist decreases and the near-closed assist increases, matching the intended effect. Across cases in which gravity affects joint motion, JODA satisfies the initial effect specification in 2.2 proposal rounds on average, showing that iteration is necessary.

5 Conclusion

We introduced JODA, a framework for synthesizing joint-level dynamics for articulated objects. Instead of modeling hidden mechanisms, it represents interaction behavior as a composition of conservative force, friction, and damping fields. An agentic generation pipeline uses a VLM to propose structured effects from visual and joint context, compiles them into physically scaled fields, and iteratively revises the proposals through diagnostic feedback. The resulting fields are executed in MuJoCo/MJX, enabling direct simulation and multiple refinement modes, including manual editing, language-guided revision, and differentiable optimization.

The current representation assumes that the dominant internal behavior can be approximated along a single joint degree of freedom with three position-dependent channels: conservative force, dry-friction magnitude, and damping. This covers many common articulated interactions, but it does not yet support more complex dynamical structures such as hidden-state effects, nonlinear damping, complex or history-dependent friction, high-dimensional DOFs, and multi-joint couplings.

Overall, JODA turns joint dynamics into an authorable and composable representation, extending articulated assets beyond geometry to executable interaction behavior. We view this as a step toward physically grounded, behavior-aware asset generation for simulation, robotics, and interactive systems. Future work includes automatically expanding the effect vocabulary, modeling multi-joint and mode-dependent dynamics, improving automatic diagnostics, and evaluating perceptual alignment with human interaction expectations.

Acknowledgments and Disclosure of Funding

We thank Liangwang Ruan, Yusen Feng, Shangbo Luan, Libin Liu, Qiyu Dai, and Baoquan Chen for their valuable suggestions on applications of this work. This work was supported by the National Natural Science Foundation of China (Project Number 62595771) and the Ant Group Research Fund. The authors declare no competing interests.

References

Matthew R. W. Brake, editor. *The Mechanics of Jointed Structures: Recent Research and Open Challenges for Developing Predictive Models for Structural Dynamics*. Springer, 2018. doi: 10.1007/978-3-319-56818-8. URL https://link.springer.com/book/10.1007/978-3-319-56818-8.

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. URDFormer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024. URL https://arxiv.org/abs/2405.11656.

Felix Endres, Jeff Trinkle, and Wolfram Burgard. Learning the dynamics of doors for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3543–3549, 2013. doi: 10.1109/IROS.2013.6696861.

Tim Engelbracht, René Zurbrügg, Matteo Wohlrapp, Martin Büchner, Abhinav Valada, Marc Pollefeys, Hermann Blum, and Zuria Bauer. Hoi!: A multimodal dataset for force-grounded, cross-view articulated manipulation. *arXiv preprint arXiv:2512.04884*, 2026. URL https://arxiv.org/abs/2512.04884.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax: A differentiable physics engine for large scale rigid body simulation. In *NeurIPS Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2106.13281.

F. N. Fritsch and J. Butland. A method for constructing local monotone piecewise cubic interpolants. *SIAM Journal on Scientific and Statistical Computing*, 5(2):300–304, June 1984. doi: 10.1137/0905021.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. URL https://arxiv.org/abs/2302.04659.

Yumeng He, Ying Jiang, Jiayin Lu, Yin Yang, and Chenfanfu Jiang. Spark: Sim-ready part-level articulated reconstruction with vlm knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. Oral presentation.

Taylor A. Howell, Simon Le Cleac'h, Jan Brudigam, Qianzhong Chen, Jiankai Sun, J. Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable physics engine for robotics. *arXiv preprint arXiv:2203.00806*, 2022. URL https://arxiv.org/abs/2203.00806.

Ruizhen Hu, Zong Yan, Jingwei Zhang, Oliver van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics*, 2017. URL https://gigl.scs.carleton.ca/papers/mobility.pdf.

Advait Jain, Hai Nguyen, Mrinal Rath, Jason Okerman, and Charles C. Kemp. The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers. In *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2010. URL https://repository.gatech.edu/entities/publication/3bda8452-1c99-48f3-b85e-89d30b3726b1.

Zhao Jin, Zhengping Che, Tao Li, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, Jing Tian, Yousong Xue, and Jian Tang. ArtVIP: Articulated digital assets of visual realism, modular interaction, and physical fidelity for

robot learning. In *International Conference on Learning Representations*, pages 75272–75301, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7a03c2bf486e56d8a25e8d5bb72ff1a2-Paper-Conference.pdf.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.13882.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martin-Martin, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. BEHAVIOR-1K: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024. URL https://arxiv.org/abs/2403.09227.

Lightwheel. Lightwheel sim-ready assets: High-quality USD assets for NVIDIA Isaac sim. GitHub repository, 2025. URL https://github.com/LightwheelAI/Lightwheel-simready-asset. 259 robotics simulation assets under CC BY-NC 4.0.

Jiayi Liu, Denys Iliash, Angel X. Chang, Manolis Savva, and Ali Mahdavi-Amiri. SINGAPO: Single image controlled generation of articulated parts in objects. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.16499.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. URL https://arxiv.org/abs/2108.10470.

Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct articulated objects via code generation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.08474.

MuJoCo Team. MuJoCo XLA (MJX) documentation, 2024. URL https://mujoco.readthedocs.io/en/latest/mjx.html.

Southco. *Hinges and Positioning Technology*, 2021. URL https://media.southco.com/media/static/documents/brochures/202108-BR-Hinges-EN.pdf.

Sugatsune. *Soft-Down and Lift Assist Stays: Lid Supports*, n.d. URL https://www.sugatsune.com/content/site-assets/Sugatsune_Resources_Catalogs/catalog-300-099-157-lid-supports.pdf.

Jan Swevers, Chris Ganseman, Dilek Bilgin Tükel, Joris De Schutter, and Hendrik Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13 (5):730–740, 1997. doi: 10.1109/70.631234.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. URL https://mujoco.org/.

Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism, 2025. URL https://arxiv.org/abs/2504.15278.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://arxiv.org/abs/2003.08515.

Kevin Zakka, Philipp Wu, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. RoboPianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2304.04150.

Mike Zhang, Yuntao Ma, Takahiro Miki, and Marco Hutter. Learning to open and traverse doors with a legged manipulator. In *Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2409.04882.

A Implementation of Composed Dynamics Fields

The three-channel representation is simulator-agnostic and can be integrated through engine-specific joint interfaces. We use MuJoCo/MJX for all reported experiments and have separately implemented and verified the representation in Isaac Sim; example Isaac Sim assets will be released on the project page.

A.1 MuJoCo-compatible asset realization

The MuJoCo model serves as the source of joint context. We extract the target joint's parent and child bodies, local and world joint axes, range, equivalent generalized inertia or mass from the mass matrix, and an engineering estimate of gravitational scale along the selected degree of freedom. This context anchors the compiler's numeric scaling and helps the VLM interpret the motion direction shown in rendered images.

A.2 Differentiable MJX implementation of dynamical fields

The composed field is executed in an MJX rollout. At each step, the PCHIP components are evaluated, and the three channels are injected into the dynamics. The conservative channel is written as an applied generalized force or torque. The damping channel updates the selected degree of freedom's damping coefficient. The friction channel updates the local dry-friction loss used by the solver. The joint-limit hint is mapped to MuJoCo/MJX limit solver parameters, such as the damping ratio of the selected joint limit model.

This implementation is designed to support differentiable optimization. Field evaluation uses JAX-compatible PCHIP interpolation, and the conservative and damping channels are directly parameterized by control-point values that can be optimized through MJX rollouts. To support differentiable fitting, we approximate the discontinuous velocity-sign term in dry friction with $\tanh(\alpha \dot{s})$, where $\dot{s} = \dot{q}/(q_{\max} - q_{\min})$ is the range-normalized joint velocity and $\alpha > 0$ controls the transition sharpness. The same field representation provides a natural parameter space for gradient-based refinement.

B Measured-interaction evaluation using Hoi!

B.1 Dataset

Hoi! [Engelbracht et al., 2026] records human interactions with articulated objects, including synchronized motion and force measurements. The main evaluation uses 48 objects featuring both opening and closing interactions, comprising 25 revolute and 23 prismatic joints across 13 environments. We select these objects from the publicly available Hoi! dataset based on the interaction measurements and annotations needed for motion recovery and simulation.

Motion and effort recovery We use the dataset's Aria pose streams, gripper force–torque measurements, calibration, static scans, and articulation and interaction annotations. Because the gripper motor coordinate describes the gripper rather than the object joint, we recover the joint coordinate q from calibrated tool-center-point (TCP) poses during handle contact. Let $\mathbf{a}$ be the opening-positive unit joint axis, $\mathbf{c}$ its origin, and $\mathbf{p}_T$ the TCP position. For a revolute joint, we unwrap the rotation of $(I - \mathbf{a}\mathbf{a}^\top)(\mathbf{p}_T - \mathbf{c})$ about the axis. For a prismatic joint, we project TCP displacement onto the axis. Sensor timestamps are aligned with the Aria poses, and calibration maps poses into the Leica world frame. We sample calibrated poses at 100 Hz across both annotated actions, preserving a shared static zero, opening sign, and angular branch. This reconstruction assumes rigid handle contact and yields a proxy rather than direct articulated-part ground truth.

We fit a robust natural cubic smoothing spline to the trusted TCP-derived coordinates within each accepted manipulation interval and obtain $q, \dot{q}, \ddot{q}$. A 45 ms smoothing scale and four robust reweighting iterations reduce tracking noise while retaining sustained motion and reversals. This yields 96 continuous curves comprising 34,542 positions across 48 objects.

The force/torque reference is derived from the dataset's gravity-compensated external wrench, low-pass filtered at 15 Hz. Calibration and synchronized TCP poses transform the wrench into the

Leica world frame, including the torque shift from the sensor to the TCP. Let $(\mathbf{f}_T, \boldsymbol{\tau}_T)$ denote this gripper-on-object wrench about the TCP. The generalized effort is $u(t) = \mathbf{a}^\top[\boldsymbol{\tau}_T + (\mathbf{p}_T - \mathbf{c}) \times \mathbf{f}_T]$ for a revolute joint (N m) and $u(t) = \mathbf{a}^\top \mathbf{f}_T$ for a prismatic joint (N), where the cross product accounts for the handle force's moment about the joint. We interpolate $u(t)$ onto the 100 Hz motion grid between valid measurements spaced at most 30 ms apart, using it as both the forward input and the inverse-effort reference.

B.2 Simulation setup

Each object uses a single-DOF MuJoCo model reconstructed from its scan and articulation annotations, with q measured in radians for revolute joints and meters for prismatic joints. Since Hoi! does not provide object masses, we assign category-based estimates from comparable objects and derive inertia and gravitational forces from these estimates and the reconstructed geometry. These parameters remain fixed throughout fitting and evaluation. All methods share the model, measured effort, initial state per run, and 0.002 s integration timestep.

The reference interval $[q_{\min}, q_{\max}]$ combines accepted scan configurations with the preliminary recovery of both actions. Each position-valid interval contributes its 1% tail quantiles, trimming at least one sample per tail and extending the bounds to include the medians of its first and last three samples.

Native MuJoCo soft stops use the smallest interval containing the reference interval and all position-valid samples from the preliminary recovery. These bounds are assumed simulation limits. Both intervals remain fixed throughout fitting and evaluation; the reference interval is not expanded to the soft-stop bounds. Predicted positions are unclipped, allowing soft-stop penetration.

Each closing rollout starts from the first observation-supported state with reliable derivatives in a continuous measured-effort interval. Recorded effort then drives the simulation without resets to subsequent reference states, including through intervals with unreliable derivatives. Recovered motion supplies only the initial state and scoring references. Motion recovery and range construction use both actions; coefficient fitting uses opening data only.

B.3 Baseline methods

Classical identification Following linear-in-parameters system identification [Swevers et al., 1997], we model measured external effort u as

$$\hat{u} = M\ddot{q} - Q_g(q) + c + k(q - q_{\mathrm{ref}}) + b\dot{q} + f_c \tanh(\alpha \dot{s}), \tag{1}$$

where M and $Q_g(q)$ are the fixed equivalent mass/inertia and generalized gravity force, and q_{ref} is the median opening coordinate; $\alpha = 20$ sets friction smoothing in normalized velocity. Using only opening samples, we form the four-column design matrix $X = [1, q - q_{\mathrm{ref}}, \dot{q}, \tanh(\alpha\dot{s})]$ and target $y = u - M\ddot{q} + Q_g(q)$. Bounded ridge least squares minimizes $\|X\beta - y\|_2^2 + 10^{-4}\|\beta\|_2^2$ for $\beta = (c, k, b, f_c)^\top$, with $b, f_c \geq 0$ and signed c, k, enforcing nonnegative dissipation. The fitted resistance terms enter the shared simulator as internal effort with the opposite sign.

Residual Gaussian process We adapt the door-dynamics GP of Endres et al. [2013] to continuously forced interactions by learning residual acceleration

$$r = \ddot{q} - \frac{u + Q_g(q)}{M}, \qquad \mathbf{x} = (s, \dot{s})^\top. \tag{2}$$

A squared-exponential kernel $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp[-\sum_j ((x_j - x'_j)/\ell_j)^2]$ uses fixed scales $(\ell_1, \ell_2) = (0.075, 1.0)$ in normalized position and velocity, plus white observation noise. Targets are standardized using opening-only statistics. An exact Cholesky GP is trained on up to 180 evenly spaced opening samples after trimming 0.1 s from each end of each contiguous interval of reliable observed opening samples; evaluation uses the respective forward and inverse scoring masks. Signal and noise magnitudes are optimized by marginal likelihood with three L-BFGS-B starts. Its posterior mean contributes internal effort $M\mu_r(s, \dot{s})$ to the shared MuJoCo model. This is a forced-residual adaptation of the original released-door experiment.

B.4 Regularized opening-only refinement

Each object's model is fitted using one opening trajectory and evaluated on a separate closing trajectory. A predominantly opening-direction trajectory provides limited excitation for separating conservative force, friction, and damping. We therefore retain the generated conservative field and fit constant additive corrections to friction and damping. Regularizing the resulting passive-effort change anchors refinement to the generated prior.

Normalization We normalize position and effort errors using the fixed scales

$$S_q = q_{\max} - q_{\min}, \qquad S_u^{\mathrm{RMS}} = \sqrt{\frac{1}{N_{\mathrm{opening}}} \sum_{j=1}^{N_{\mathrm{opening}}} u_j^2}.$$

Here S_q is the reference coordinate span, and S_u^{RMS} uses the object's N_{opening} quality-controlled opening reference samples from the preliminary recovery.

Parameterization For each object, we optimize $\theta = (\theta_d, \theta_f)^\top$, where θ_d and θ_f are dimensionless coefficients for constant damping and friction offsets:

$$\Delta d = \frac{S_u^{\mathrm{RMS}}}{S_v}\theta_d, \qquad \Delta f = S_u^{\mathrm{RMS}}\theta_f.$$

Here S_v is the median of reliable opening speeds at least $0.01S_q$/s, or this threshold if none qualify. For generated fields $d_0(q)$ and $f_0(q)$, the refined fields are $d_\theta(q) = \max\{0, d_0(q) + \Delta d\}$ and $f_\theta(q) = \max\{0, f_0(q) + \Delta f\}$. Each offset is shared across the six PCHIP knots at $s = 0, 0.2, \ldots, 1$.

Objective For the N reliable opening fitting samples, we omit the recording index e and define normalized rollout and inverse-effort residuals

$$\epsilon_i = \frac{\hat{q}_i - q_i}{S_q}, \qquad r_i = \frac{\hat{u}_i^{\mathrm{inv,free}} - u_i}{S_u^{\mathrm{RMS}}}.$$

Here $\hat{u}_i^{\mathrm{inv,free}}$ is MuJoCo inverse effort at the recovered $q, \dot{q}, \ddot{q}$ with joint-limit reactions disabled. We use $\lambda_u = 1$ for all 48 objects, giving equal weight to normalized trajectory and inverse-effort errors, and minimize

$$\begin{aligned}\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(\epsilon_i^2 + \lambda_u r_i^2) \\ + 0.1\langle \Delta p^2 \rangle_G + 0.03\langle (D_2 \Delta p)^2 \rangle_H + 3\langle \Delta p^2 \rangle_{G_U} + 10^{-4}\langle \theta^2 \rangle.\end{aligned} \tag{3}$$

Here Δp is the change from the generated profile in net passive effort after nonnegative clipping, divided by S_u^{RMS}; brackets denote means over the indicated entries. The four regularizers penalize effort change, curvature, changes outside opening support, and coefficient magnitude.

Regularization grid The fixed grid G combines 41 equally spaced positions across $[q_{\min}, q_{\max}]$ with six velocities: both signs of the 25th, 50th, and 90th percentiles of reliable opening speeds at least $0.01S_q$/s. This threshold also floors the speed values. D_2 is the second position difference divided by $(1/40)^2$, evaluated at the 39 interior nodes and six speeds (H). G_U contains grid entries outside the reliable opening position interval; its penalty is zero when empty.

Optimization Both components of θ are bounded to $[-10, 10]$ and initialized at zero. We use trust-region reflective least squares over MuJoCo rollouts, with Jacobian scaling, a 10^{-6} termination tolerance, and at most 200 function evaluations; nonconverged fits are rejected. Each object's coefficients are fitted using only its opening trajectory. The fitted corrections are applied in full and frozen for closing evaluation.

Table 2: Per-object closing errors for all 48 objects in the Hoi! evaluation. Lower NRMSE is better. Classical denotes classical identification, GP denotes the residual GP, and Zero-shot and Optimized denote the JODA variants. R/P indicates a revolute/prismatic joint. Bold marks the lowest displayed error for each object and metric.

Object	Joint	Trajectory NRMSE				Inverse-effort NRMSE			
		Classical	GP	Zero-shot	Optimized	Classical	GP	Zero-shot	Optimized
bathroom_2:2	P	**0.2521**	0.6621	0.3517	0.3938	0.6794	2.3882	**0.5255**	0.5319
bedroom_4:1	R	0.5519	**0.0768**	0.2863	0.4414	0.8906	1.0802	**0.6190**	0.6221
bedroom_5:1	P	**0.0386**	0.1629	0.1510	0.0995	**0.6077**	16.0979	0.9717	0.8691
bedroom_5:2	P	**0.0733**	0.2699	0.1232	0.1165	**0.6148**	1.7421	0.7669	0.7321
bedroom_5:3	P	**0.0748**	0.3841	0.1480	0.1402	0.7536	5.7940	0.7701	**0.7149**
bedroom_5:4	R	0.1589	0.5310	0.1565	**0.1289**	0.7159	4.1624	0.6156	**0.5840**
bedroom_5:5	R	0.8222	0.0925	0.0534	**0.0255**	1.7580	1.0275	0.8854	**0.8764**
bedroom_6:1	P	0.1152	0.5785	0.0811	**0.0527**	**1.1211**	6.3550	1.4089	1.3299
bedroom_6:2	P	0.2419	0.6357	0.1985	**0.0258**	1.3451	3.2875	1.3893	**1.1539**
bedroom_8:1	R	0.7669	0.2103	0.1658	**0.0698**	1.0551	1.0586	0.2701	**0.2317**
bedroom_8:3	R	0.7804	0.7624	0.1439	**0.0412**	0.6321	0.6461	0.2866	**0.2681**
bedroom_8:4	R	0.6997	**0.1178**	0.2721	0.1591	2.7589	1.2223	**0.1760**	0.1993
bedroom_8:5	R	0.7579	**0.1624**	0.4961	0.2255	1.5382	1.2610	**0.1811**	0.2013
bedroom_8:6	R	0.7166	0.4107	0.5809	**0.3062**	0.7027	0.9553	0.3818	**0.3399**
kitchen_7:2	P	0.3189	0.7491	0.3010	**0.2722**	**0.2236**	10.5070	0.2520	0.2484
kitchen_14:2	P	0.6663	0.2337	**0.0464**	0.0466	2.1253	1.6155	0.4973	**0.4972**
kitchen_14:3	R	0.7485	0.5257	0.4081	**0.3060**	0.8692	2.3638	0.4462	**0.4243**
kitchen_14:6	P	0.7431	0.7568	0.0948	**0.0473**	0.9666	10.2039	0.8216	**0.7897**
kitchen_14:10	R	0.7446	0.7533	0.1935	**0.1405**	0.7827	3.4072	0.7624	**0.7095**
kitchen_14:11	R	0.7092	**0.1810**	0.2740	0.3291	0.8996	0.9933	0.5566	**0.5242**
kitchen_14:12	R	0.4141	0.4152	0.2219	**0.1586**	1.0797	**0.8338**	0.8992	0.8553
kitchen_14:13	R	0.3451	0.1763	**0.0777**	0.1319	1.0511	1.7506	0.6208	**0.5611**
kitchen_15:1	P	**0.0612**	0.7124	0.0973	0.0944	**0.9910**	2.7176	0.9993	0.9959
kitchen_15:7	P	0.2797	0.6799	0.2304	**0.2015**	1.1076	1.6736	0.4610	**0.4067**
kitchen_15:10	R	0.3928	**0.1275**	0.1648	0.1381	**1.1246**	1.2832	1.2622	1.1739
kitchen_15:11	R	0.4766	**0.1428**	0.1894	0.1538	1.2748	1.4296	1.1847	**1.1445**
kitchen_17:1	P	0.2347	0.6783	**0.2054**	**0.2054**	0.6237	4.9802	**0.5771**	**0.5771**
kitchen_17:6	R	0.3480	0.6659	**0.1986**	0.6336	1.3683	3.7164	0.8656	**0.8576**
kitchen_17:11	R	0.1458	0.2966	0.2641	**0.1079**	0.7137	8.0735	0.3616	**0.2800**
kitchen_17:12	R	**0.1286**	0.3579	0.2429	0.2098	0.8841	0.9524	0.4294	**0.3739**
kitchen_17:13	R	0.7532	0.1407	0.1066	**0.0742**	1.9382	0.8274	0.6286	**0.5749**
livingroom_1:1	R	0.7072	**0.1911**	0.5031	0.5271	0.9759	**0.5866**	0.7433	0.7952
livingroom_1:5	R	0.7175	**0.0835**	0.6941	0.6840	1.2809	**0.5380**	0.6992	0.7439
livingroom_1:6	R	0.7019	**0.2119**	0.3422	0.4052	1.1152	0.6848	**0.5425**	0.5690
office_1:8	P	0.7478	0.7480	**0.0895**	0.1773	1.7272	6.2078	0.4267	**0.3807**
wardrobe_1:4	P	0.3016	0.5908	**0.0341**	0.1211	0.7762	5.9588	0.4299	**0.4252**
wardrobe_1:7	P	**0.0879**	0.3719	0.2446	0.2262	**0.4604**	8.1322	0.7564	0.6349
wardrobe_1:8	R	0.6603	0.2509	**0.0228**	0.2507	1.1309	1.8067	0.2787	**0.2726**
wardrobe_1:10	R	0.6344	**0.1926**	0.2084	0.3919	1.2802	0.7669	**0.3155**	0.6867
wardrobe_1:11	P	**0.0684**	0.2810	0.2396	0.1579	**0.9337**	10.0044	1.7200	1.2972
wardrobe_1:12	R	0.0846	**0.0843**	0.3974	0.2382	0.6030	1.0408	0.6010	**0.5749**
wardrobe_1:14	P	0.1331	0.6945	**0.0328**	0.0532	0.6008	3.4104	0.5895	**0.5647**
wardrobe_1:15	P	0.2506	0.5781	0.0878	**0.0211**	0.6469	4.9460	0.5319	**0.5178**
wardrobe_1:16	P	**0.0551**	0.7141	0.3023	0.3349	**1.5699**	9.7568	1.7815	1.7794
wardrobe_2:1	P	**0.0487**	0.7017	0.1948	0.2397	**0.9933**	3.0959	1.0849	1.0748
wardrobe_2:2	P	**0.0997**	0.6488	0.1388	0.1859	**0.7238**	2.7050	0.8280	0.8166
wardrobe_2:3	P	0.1289	0.6877	**0.0434**	0.0621	0.8855	1.3264	0.8485	**0.7896**
wardrobe_2:4	P	0.4554	0.5685	0.4144	**0.3957**	1.1024	1.6772	0.8849	**0.7644**
Mean (48 objects)		0.4051	0.4219	0.2191	**0.2073**	1.0417	3.4802	0.7070	**0.6737**

B.5 Evaluation metrics and results

Metrics We use two complementary metrics to evaluate closing interactions. *Trajectory NRMSE* measures open-loop motion prediction under the recorded effort:

$$\mathrm{NRMSE}_e = \frac{1}{S_q}\sqrt{\frac{1}{N_e^q}\sum_{i=1}^{N_e^q}(\hat{q}_{e,i} - q_{e,i})^2}. \tag{4}$$

Here $q_{e,i}$ is the recovered position defined above, and $\hat{q}_{e,i}$ is the simulated position at the same sample time and in the same coordinate frame. The N_e^q samples are observation-supported positions in closing recording e. *Inverse-effort NRMSE* measures error in the external effort required to reproduce the recovered motion:

$$\mathrm{NRMSE}_e^u = \frac{1}{S_u^{\mathrm{RMS}}}\sqrt{\frac{1}{N_e^u}\sum_{i=1}^{N_e^u}(\hat{u}_{e,i}^{\mathrm{inv}} - u_{e,i})^2}. \tag{5}$$

Here $\hat{u}_{e,i}^{\mathrm{inv}}$ is full MuJoCo inverse effort at the shared recovered $q, \dot{q}, \ddot{q}$, and $u_{e,i}$ is measured effort. The N_e^u samples require observation-supported positions, reliable derivatives, and valid effort. Both metrics use the fixed normalization scales defined for refinement. The modeled stops exert no reaction at inverse-scored states.

We average episode scores within each object and action, then equally across objects. Both metrics are dimensionless, allowing revolute and prismatic joints to be pooled.

Results Table 1 shows that JODA zero-shot achieves lower mean errors than both fitted baselines for each joint type and metric, supporting the value of the generated dynamics prior for predicting measured interactions. Regularized refinement yields the lowest mean errors across both metrics and joint types, reducing overall trajectory and inverse-effort NRMSE by 5.4% and 4.7%, respectively, relative to JODA zero-shot. The trajectory improvement is more pronounced for revolute joints, while the inverse-effort improvement is larger for prismatic joints.

The GP's lower mean revolute trajectory error relative to classical identification is accompanied by higher inverse-effort error, highlighting the importance of evaluating both motion and required effort. Relative to JODA zero-shot, optimized JODA lowers trajectory NRMSE on 30 objects and raises it on 18; inverse-effort NRMSE decreases on 39 objects and increases on nine. Table 2 reports both metrics for every object and all four methods.

C VLM Prompt and Effect Proposal Specification

We design a structured prompting interface that enables the VLM to infer joint-level effects in a physically grounded and compositional form. Given joint context, multi-state rendered images, and effect-template previews, the model infers equivalent joint-level effects across the full joint range.

C.1 Prompt Excerpt

We provide a shortened excerpt of the prompt used for VLM-based effect proposal in Fig. 12. The full prompt is provided in the supplementary material. The prompt casts the model as a structured mechanics analyst and enforces physically grounded reasoning and structured output.

C.2 Output JSON Schema (Compact)

The VLM outputs a structured JSON object describing joint-level dynamical effects, which is summarized in Listing 1. The full JSON schema is provided in the supplementary material. This schema separates semantic effect specification from numerical realization. Qualitative strength labels are later mapped to physically scaled parameters by the compiler.

Listing 1: **VLM Output JSON Schema.** The structured output separates semantic effect specification from numerical realization.

```
{
  "joint_summary": {
    "joint_name": str,
    "joint_type": "revolute | prismatic",
    "motion_description": str,
    "overall_confidence": float
  },

  "whole_motion_descrptn": str,
  "gravity_can_be_ignored": bool,

  "joint_limit_hint": {
    "selected_side": "low_end | high_end | none",
    "elasticity": "none | weak | medium | strong"
  },

  "effect_proposals": [
    {
      "effect_name": str,                // from template library
      "start_ratio": float,              // in [0, 1]
      "end_ratio": float,                // in [0, 1]

      "strength": {
        "conservative": "none | weak | medium | strong | dominant",
        "friction":     "none | weak | medium | strong | dominant",
        "damping":      "none | weak | medium | strong | dominant"
      },

      "refineFactor": {
        "conservative": float,
        "friction": float,
        "damping": float
      },

      "confidence": float,
      "reason": str
    }
  ]
}
```

Table 3: Predefined effect templates defining the joint-level behavior vocabulary.

Template	**Behavior description**
constant_friction_hinge	Constant dry friction over the interval
constant_damping_hinge	Constant velocity-dependent damping
constant_positive_conservative_hinge	Constant drive toward increasing q
constant_negative_conservative_hinge	Constant drive toward decreasing q
detent_internal	Local click-stop with interior equilibrium
bistable_mechanism	Two stable regions separated by a barrier
bistable_mechanism_internal	Two interior stable points
magnetic_return_to_low_end	Increasing attraction toward low end
magnetic_return_to_high_end	Increasing attraction toward high end
spring_return_to_low_end	Spring-like return toward low end
spring_return_to_high_end	Spring-like return toward high end
spring_loaded_snap_detent_to_low_end	Snap-in latch near low end
spring_loaded_snap_detent_to_high_end	Snap-in latch near high end

Prompt Design

Role. You are a careful multimodal mechanics analyst working in a high-precision pipeline. Your output will be used as structured input to a downstream compiler. Prioritize correctness, consistency, and physical plausibility over free-form creativity.

Task. Given one articulated joint, infer the equivalent joint-level effects that describe the felt behavior along the entire motion range. These effects represent observable interaction behavior, not explicit claims about hidden mechanisms.

Inputs.

- Joint context (type, axis, range, gravity relation);
- multi-state rendered images along the joint coordinate;
- template previews showing reference force, friction, and damping profiles.

Output. Return a JSON object (detailed in App. C.2) describing:

- a whole-motion description,
- a gravity-importance judgment,
- a joint-limit hint,
- a set of effect proposals (type, interval, qualitative strength).

Constraints.

- Use only predefined effect templates (summarized in Table 3).
- Represent behavior as equivalent joint-level effects.
- Reason over the full motion range.
- Prefer simple explanations when multiple effects are plausible.

Iterative refinement. When provided with a rendered feedback result, either return `complete` if the behavior matches the intent, or revise the effect proposals to better match the desired motion. The full prompt is provided in the supplementary material.

Figure 12: Prompt design for multimodal mechanics analysis.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We confirm that the main claims in the abstract and introduction reflect the paper's contribution and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The proposed method is described in detail in Sec. 3, and the experiment setup is detailed in Sec. 4 as well as App. A to ensure reproducibility. Additional details are given in the supplementary.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code will be released publicly on the project page upon publication.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The experiment setup is detailed in Sec. 4 as well as App. A to ensure reproducibility. Additional details are provided in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report quantitative errors for two measured force profiles and 48 objects from the Hoi! dataset. Per-object results are provided in App. B, but we do not report error bars or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiment setup is given in Sec. 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirmed that the research conducted in the current paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper discusses applications and limitations in Section 5, but does not explicitly discuss broader societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not introduce or release large-scale pretrained models, datasets, or generative systems with inherent high risk of misuse. The method focuses on structured simulation modeling of articulated object dynamics. While it may improve realism in simulated environments, it does not directly enable high-risk applications that would require specific safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite PartNet-Mobility Xiang et al. [2020], Hoi! Engelbracht et al. [2026], and Lightwheel Sim-Ready Assets Lightwheel [2025] in Sec. 4. PartNet-Mobility is subject to non-commercial research terms; Hoi! is licensed under CC BY 4.0, and Lightwheel assets under CC BY-NC 4.0.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and example assets built on existing datasets will be released with documentation on the project page upon publication.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work uses the existing Hoi! dataset and does not conduct new crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work uses the existing Hoi! dataset and does not conduct new human-subject studies requiring IRB approval.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed method uses a vision-language model (VLM) as a core component to infer structured joint-level dynamical effects from visual observations and joint context. This usage is described in the main paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.