

ForceTwin: Physics-informed Digital Twins for Robotic Manipulation from Instrumented Human Interaction

Tim Engelbracht^{1,†} René Zurbrügg¹ Mayank Mittal^{1,2} Marco Hutter¹
 Marc Pollefeys^{1,3} Hermann Blum⁴ Zuria Bauer¹

¹ETH Zurich ²NVIDIA ³Microsoft ⁴University of Bonn

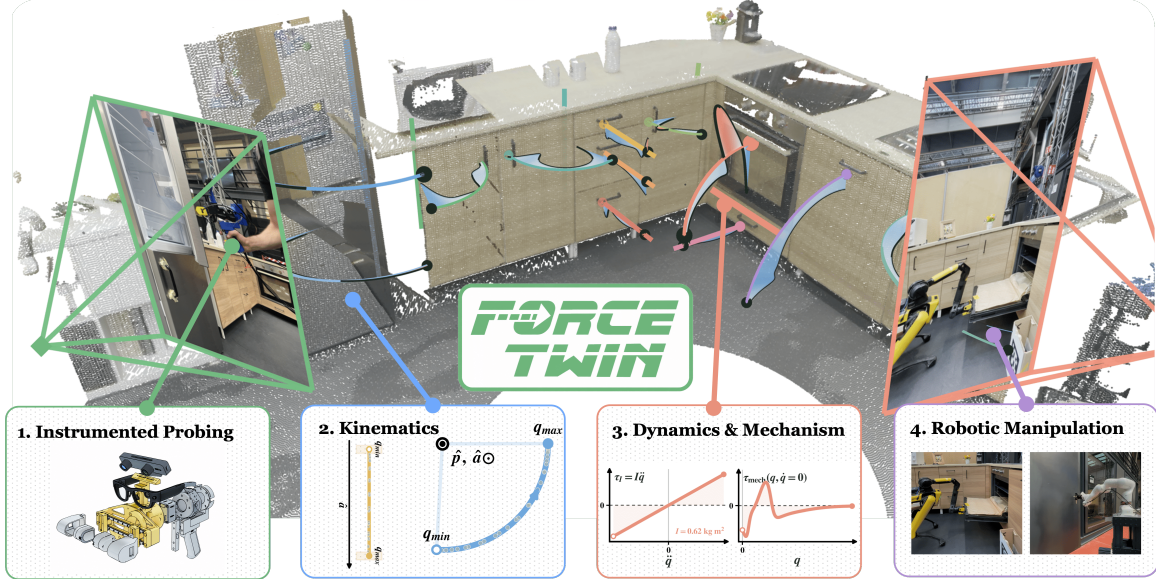


Fig. 1. FORCE TWIN builds a physics-informed digital twin from instrumented human interaction. Interaction trajectories estimate object kinematics, while measured wrenches identify dynamics and mechanism behavior. Combined with semantic 3D scene reconstruction, FORCE TWIN supports dynamics-aware robot control and high-fidelity simulation parameters for reinforcement learning.

Abstract—Manipulating objects requires understanding not only their motion, but also the physical properties that determine it. For articulated objects, these include inertia, friction, and mechanisms such as springs or door closers, whose effects can vary with configuration and velocity. Such properties are not directly observable from appearance: visually identical doors may require very different effort to manipulate. Existing digital-twin pipelines recover primarily kinematics or assign static physical parameters from visual and language priors, which can yield physically implausible estimates. As a result, state-dependent mechanism dynamics remain unidentified and are not represented in standard asset formats. We present FORCE TWIN, a system for identifying physics-informed digital twins of articulated objects from instrumented human interaction. A person probes an object using a handheld force-sensing gripper, providing synchronized poses and interaction forces from which we estimate the articulation, parametric dynamics including inertia, Coulomb friction, viscous damping, and a structured neural residual capturing state-dependent mechanism forces. FORCE TWIN nearly halves the inertial-parameter error of a VLM prior. As a feedforward dynamics model for impedance control on a Spot and a Franka FR3, FORCE TWIN achieves 87% goal completion across nine object-embodiment pairs, compared with 60% using VLM-prior and 57% using kinematics-only twins, with the largest gains on

objects whose strong mechanisms cause both baselines to stall. We further use the identified twins to train whole-body door-traversal policies and deploy them in the real world. Project page: timengelbracht.github.io/forcetwin-website

I. INTRODUCTION

Robotic manipulation requires more than knowing how an object can move. To open a drawer, a dishwasher, or a door with a closer, a robot must also account for the forces induced by inertia, friction, and internal mechanisms. These dynamics can vary substantially between visually similar objects and are often impossible to infer from appearance alone: the same drawer may be empty or heavily loaded, while a concealed closer can make an otherwise ordinary door strongly resist motion. A digital twin for manipulation must therefore capture not only an object’s geometry and articulation, but also the instance-specific dynamics that govern its interaction.

These dynamics have to be measured. Appearance provides at best a prior, while motion alone identifies inertia, friction, and mechanism torque only up to a common scale that the contact wrench resolves. Feedback cannot simply replace the missing model: strong resistance may indicate that a drawer is stuck and the robot should stop, or be

[†]Corresponding author: tengelbracht@ethz.ch

the expected behavior of a spring-loaded door that requires additional effort.

Existing twin pipelines neither measure this effort nor model its state-dependent component. Reconstruction methods recover kinematic structure [1]–[4], while recent methods assign physical parameters from visual and language priors [5], [6]. Robot-probing approaches [7]–[9] measure interaction, but recover at most constant friction models and require robot deployment. None identify the nonlinear, state-dependent response of mechanisms such as door closers or spring-loaded catches, which no set of constant parameters can describe, and standard asset formats provide no representation for this response.

We take a different route (Fig. 2): a person measures the object. Handheld force-sensing grippers have made in-the-wild interaction capture practical [10], [11]; we repurpose them for system identification. With FORCETWIN, a person probes an articulated object, and from the tool trajectory and contact wrenches we identify the object’s kinematics and dynamics (Fig. 1), decoupling identification from robot deployment. Humans can reach objects without a robot present, safely excite their full range across varied speeds, and do so faster than contact-based robot exploration. We model joint effort semi-parametrically. Inertia, Coulomb friction, and viscous damping are estimated under nonnegativity constraints, yielding physically valid parameters that map directly to simulators and controllers. A structured neural residual captures the remaining state-dependent mechanism response. The resulting twin can be used directly for model-based control or exported as a simulation asset.

In summary, our contributions are:

- We introduce FORCETWIN, which identifies instance-specific articulated-object dynamics from instrumented human interaction, including nonlinear, state-dependent mechanism responses.
- We formulate identification as a semi-parametric partially linear model, combining physically constrained inertia and friction parameters with a structured neural residual, yielding simulation-compatible parameters and a predictive model of joint effort.
- We show that one identified twin serves both control and simulation. As the feedforward model of an impedance controller, it reaches 87% goal completion across nine object–embodiment pairs, where prior-based and kinematics-only twins reach 60% and 57% and stall entirely on objects with strong mechanisms; incorporated into simulation assets, the identified dynamics support policy learning, where we train whole-body door-traversal policies and deploy them in the real world.

II. RELATED WORK

Articulated Object Perception and Kinematics. Estimating articulation models by acting on the world is a classic instance of interactive perception [12]: a robot pushes an

object and infers its kinematic model from the observed motion [7], [13]. Later work removes the interaction, estimating articulation from a single RGB-D frame [14], from two observed object states [15], with learned category-independent models [16], or from in-the-wild video [17], and scene-graph pipelines lift such estimates to scene scale for robotic manipulation [1], [2], [4], [18]. All of these methods estimate kinematic structure, i.e., joint types and axes, but not the dynamic parameters that govern interaction forces; our kinematics stage builds on this line of work, adopting the twist formulation of works such as [16], [19] and interactive BIC-based model selection of [13], but treats the kinematic model as the input to dynamic identification rather than the final result.

Instrumented Interaction and Physical Model Identification. Closest to our setting are methods that identify physical parameters from forceful interaction. Door inertia and a velocity-dependent deceleration profile have been identified from robot interaction and released-door trajectories [20], and human opening forces on doors and drawers have been captured with an instrumented hook, yielding force-versus-angle profiles used for haptic recognition and monitoring [21]. We share their premise that dynamics must be measured, but differ in what is recovered and for what: our semiparametric formulation yields interpretable, simulation-ready parameters plus a nonlinear mechanism model, rather than a per-object regression, and our capture requires only a handheld gripper rather than a robot. Robot-mounted estimators also go beyond kinematics: coupled recursive filters over vision, wrist forces, and proprioception additionally estimate a quasi-static Coulomb friction model, i.e., stiction and constant kinetic friction, while deliberately neglecting inertial and viscous effects [8], and factor-graph estimators use force factors to disambiguate the articulation but identify no dynamic parameters [9]. Our identification retains all of these terms, inertia, Coulomb and viscous damping, under physical constraints, and adds the nonlinear mechanism response. Ditto [22] builds “digital twins of articulated objects from interaction,” but its twins carry default physical parameters, while the dynamics parameters are not identified. Finally, handheld instrumented grippers [10], [11], [23] have made in-the-wild interaction capture practical, so far primarily to record demonstrations for policy learning; we repurpose this hardware for system identification.

Real-to-Sim and Simulation Asset Creation. A complementary line of work turns real scenes into simulation-ready articulated assets: from images alone [24], from a robot push [25], through human-in-the-loop annotation [26], or by retrieving similar “digital cousins” from an asset database [27]. In all four, the resulting physics is defaulted, annotated, or inherited from the retrieved asset. A parallel thread assigns physical parameters from visual and language priors [3], [5], [6], fits them to visual tracking [28], or tunes simulator distributions against real task rollouts [29]. None of these approaches measure the interaction forces of the specific instance being modeled. FORCETWIN complements this line of work with per-instance dynamics, identified from

force measurements and exported in a format such pipelines can consume.

III. METHOD

A. Problem Formulation

We consider a rigid articulated part with a single degree of freedom and identify its kinematics and interaction dynamics from instrumented human probing. The observations consist of two synchronized streams,

$$\mathcal{D} = \{ {}^W T_{\text{TCP},i}, {}^W \mathcal{F}_i \}_{i=1}^N, \quad (1)$$

where ${}^W T_{\text{TCP},i}$ is the pose of the tool center point (TCP) in a world frame W , and ${}^W \mathcal{F}_i \in \mathbb{R}^6$ is the contact wrench applied by the tool to the object after compensating for the weight and inertial loading of the instrumented gripper. We assume that the tool maintains a rigid, non-slip grasp and that the articulated part follows an ideal revolute or prismatic joint during the retained measurements. Unless marked otherwise, all quantities are expressed in W , and bold denotes a stacked multi-component quantity. From $\mathcal{D}$, we estimate the articulation class $\kappa \in \{\text{revolute}, \text{prismatic}\}$, the joint twist ${}^W \xi$, the observed range $\mathcal{Q} = [q_{\min}, q_{\max}]$, and a generalized-effort model $\hat{\tau} = \hat{\tau}(q, \dot{q}, \ddot{q})$. We represent the latter by simulator-compatible effective parametric coefficients $\beta = [\mu, b, I]^\top$ and a structured mechanism residual $\mathcal{M}_{\text{mech}} = \{g_\theta, c_\phi\}$. Together, these quantities form the physical model

$$\mathcal{M}_{\text{phys}} = (\kappa, {}^W \xi, \mathcal{Q}, \beta, \mathcal{M}_{\text{mech}}). \quad (2)$$

The model is estimated independently of a particular geometric reconstruction. A physics-informed digital twin is obtained afterwards by registering and associating $\mathcal{M}_{\text{phys}}$ with a geometric asset or a robot's object representation.

B. Kinematics Estimation

Under the rigid-grasp assumption, the tracked TCP positions $\{p_i\}_{i=1}^N$ follow the one-dimensional motion of the articulated part. We represent this motion using a screw in product-of-exponentials form [13], [16], [30],

$$\bar{p}_i = \exp([{}^W \xi]^\wedge q_i) \bar{p}_0, \quad (3)$$

where $\bar{p}_i \in \mathbb{R}^4$ denotes the homogeneous TCP position, q_i is the scalar joint coordinate, and ${}^W \xi = [\omega^\top, v^\top]^\top \in \mathbb{R}^6$ is the joint twist, and $[\cdot]^\wedge$ maps a twist to its 4×4 matrix form in $\mathfrak{se}(3)$. We remove the scale ambiguity between ${}^W \xi$ and q by imposing $\|\omega\|_2 = 1$ for revolute joints and $\|v\|_2 = 1$ for prismatic joints. We set the coordinate origin at the first retained configuration and orient the axis so that the first sustained displacement has positive q . These conventions fix the units and signs of the subsequently identified dynamic parameters. A revolute candidate satisfies $\omega \neq 0$ and $\omega^\top v = 0$, while a prismatic candidate satisfies $\omega = 0$. We fit both candidates and select the articulation class using the Bayesian information criterion [13], [31]. We then refine the selected model in a factor graph, similar to [9], jointly estimating the joint geometry and the per-sample states $\{q_i\}$. For revolute joints, the refinement enforces zero screw pitch. The observed extrema of the refined states define $\mathcal{Q} = [q_{\min}, q_{\max}]$.

C. Dynamics Identification

Wrench Compensation and Generalized Effort. The force-torque sensor measures both the contact wrench and the wrench generated by the instrumented gripper itself. We transform the measured wrench into the world frame and subtract the calibrated gravitational and inertial wrench of the gripper, yielding the estimated contact wrench ${}^W \mathcal{F}_i$ applied by the tool to the object. We order spatial twists as angular followed by linear velocity and wrenches as moment followed by force. The kinematic estimate then defines the object Jacobian at the TCP,

$$J_{\text{obj}}(q_i) = \begin{cases} [\omega \times (p_i - o), \omega]^\top, & \kappa = \text{revolute}, \\ [v, \mathbf{0}]^\top, & \kappa = \text{prismatic}. \end{cases} \quad (4)$$

where o is a point on the revolute axis, and J_p denotes the linear block of J_{obj} , the three rows that map $\dot{q}$ to the TCP velocity. By virtual work, the generalized effort applied by the tool is $\tau_i = J_{\text{obj}}(q_i)^\top {}^W \mathcal{F}_i$. Under the rigid-grasp and ideal-joint assumptions, wrench components that perform no virtual work along the permitted articulation lie in the null space of $J_{\text{obj}}^\top$ and do not contribute to τ_i . The projection does not remove errors caused by wrench calibration, joint-axis estimation, grasp compliance, or slip; these effects remain part of the measurement residual. We obtain $\dot{q}_i$ and $\ddot{q}_i$ by differentiating a smoothed joint-state trajectory. Samples near joint limits, with negligible velocity, or with implausibly large differentiated accelerations are removed using excitation gates.

Generalized-Effort Model. We define τ_i as the generalized effort applied to the object and model the effort required to realize the observed motion as

$$\tau = \underbrace{I\ddot{q} + \mu \operatorname{sgn}(\dot{q}) + b\dot{q}}_{\text{effective parametric dynamics}} + \underbrace{g_\theta(q) + c_\phi(q, |\dot{q}|)\dot{q}}_{\text{structured mechanism residual}} + \varepsilon, \quad (5)$$

where I denotes rotational inertia for revolute joints and effective mass for prismatic joints, μ is the Coulomb friction magnitude, b is viscous damping, and ε contains measurement noise and unmodeled effects. We write $\tau_{\text{mech}}(q, \dot{q}) = g_\theta(q) + c_\phi(q, |\dot{q}|)\dot{q}$ for the structured residual as a whole.

The sign convention in Eq. (5) describes the effort that must be applied to move the object. Consequently, the compensating damping effort is $+c_\phi\dot{q}$. The corresponding damping effort exerted by the object is $-c_\phi\dot{q}$, whose mechanical power is

$$P_{\text{damp}} = -c_\phi(q, |\dot{q}|)\dot{q}^2 \leq 0 \quad (6)$$

when $c_\phi \geq 0$. The function $g_\theta(q)$ captures configuration-dependent generalized loads. For joints whose motion is orthogonal to gravity, this term primarily represents mechanisms such as door closers and spring-loaded elements. For a horizontal revolute axis, it may also contain gravitational torque unless gravity is compensated using an independently available mass and center-of-mass estimate. We therefore interpret g_θ as a configuration-dependent load rather than claiming that it uniquely isolates a mechanical spring.

Residual Fitting. We first estimate the effective parametric coefficients using samples with sufficient motion and force

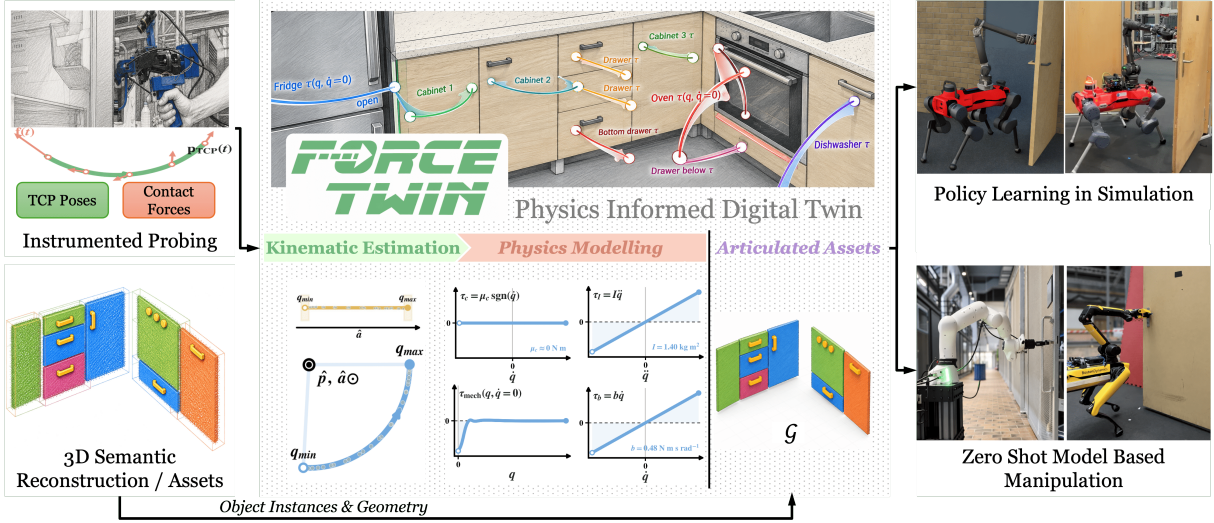


Fig. 2: **The FORCETWIN pipeline.** Instrumented human interaction provides synchronized tool poses and contact wrenches from which we identify an articulation model and its generalized-effort dynamics. Geometry is reconstructed from the same recording or supplied independently, and the physical model is registered to it to produce a twin for robot control or simulation.

excitation. For sample i , define

$$z_i = [\text{sgn}(\dot{q}_i) \quad \dot{q}_i \quad \ddot{q}_i]^\top, \quad \beta = [\mu \quad b \quad I]^\top. \quad (7)$$

Stacking the retained regressors into Z gives

$$\hat{\beta} = \underset{\beta \geq 0}{\text{argmin}} \|\tau - Z\beta\|_2^2. \quad (8)$$

The elementwise nonnegativity constraint enforces nonnegative effective inertia, Coulomb friction, and viscous damping. These quantities correspond to properties commonly supported by rigid-body simulators, although their exact interpretation and implementation remain backend-dependent. We next fit the structured nonlinear model to the residual

$$r_i = \tau_i - z_i^\top \hat{\beta}, \quad (9)$$

using $\hat{r}_i = \tau_{\text{mech}}(q_i, \dot{q}_i)$, $c_\phi \geq 0$. The configuration-dependent head g_θ is sign-unconstrained and takes only q as input. The damping coefficient is produced by softplus outputs, with separate heads for positive and negative motion, allowing directional asymmetry while preserving nonnegative damping. The resulting model is partially linear, but its decomposition is not generally unique. In particular, a constant component of c_ϕ is indistinguishable from the linear damping coefficient b , and configuration-dependent gravity is indistinguishable from a mechanical load represented by g_θ . Correlations between q , $\dot{q}$, and $\ddot{q}$ can introduce further coupling between the fitted terms. We therefore interpret $\hat{\beta}$ as effective simulator-compatible parameters and evaluate the predictive fidelity of the complete effort model. The structured residual and probing protocol regularize the decomposition, but do not make every component uniquely identifiable. We fit the model sequentially: nonnegative least squares followed by residual-network training. Training minimizes residual MSE using full-batch AdamW (learning rate and weight decay 10^{-3}), with a random 75/25 training/validation split, at most 1200 epochs, and early stopping after 120 epochs without validation improvement; we retain the best-validation weights for the mechanism term.

D. Twin Assembly

The identified physical model is expressed in the probing frame W and joint coordinate defined above. To attach it to a target geometric representation with frame A , we first register the probing frame using ${}^A T_W$. The joint twist transforms according to ${}^A \xi = \text{Ad}_{{}^A T_W} {}^W \xi$. Rigid frame changes preserve the scalar joint coordinate and generalized effort. If the target asset instead uses a different coordinate convention

$$q_A = s q + q_{\text{off}}, \quad s \in \{-1, 1\}, \quad (10)$$

we transform the valid range and reparameterize the mechanism model. For the complete mechanism effort $\tau_{\text{mech}}(q, \dot{q})$, the target convention uses $\tau_{\text{mech}}^A(q_A, \dot{q}_A) = s \tau_{\text{mech}}(s(q_A - q_{\text{off}}), s\dot{q}_A)$. The nonnegative magnitudes I , μ , and b are unchanged under the sign reversal, while their associated signed efforts follow the transformed joint coordinate. We then associate the registered model with the corresponding articulated instance, as illustrated in Fig. 3. The parametric dynamics are mapped to the corresponding joint or actuator properties provided by the target backend. When the backend supports runtime joint-effort evaluation, the nonlinear model is evaluated directly. For backends that cannot execute the learned network, we approximate it with a backend-supported spring-damper representation during export. This conversion is distinct from physical identification and may introduce an additional approximation. In our experiments, the geometric representation is reconstructed from the same recording, so it already shares the probing frame. We associate the manipulated handle with the tracked TCP and use the reconstructed functional relation to identify the corresponding panel. Other geometric backends or existing assets require only a registration and an instance-association procedure. The resulting twin can be consumed either as a feedforward dynamics model for robot control or as an articulated simulation asset.

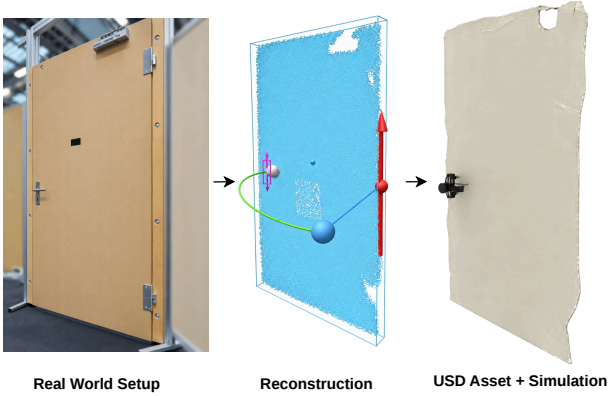


Fig. 3: **Construction of a physics-informed twin.** (a) Reconstructed panel instance, associated handle, and registered joint axis. (b) Exported articulated twin. The physical model is transformed into the asset frame, reparameterized to match its joint-coordinate convention, and attached to the corresponding articulated instance.

IV. EXPERIMENTS

Evaluation proceeds along two axes. First, we ask whether the estimated parameters are accurate and whether that accuracy justifies the model’s complexity: Section IV-B measures fidelity against ground truth inertia measurements and real-world behavior, and Section IV-C investigates the contribution of each model component. Second, we investigate the utility of ForceTwin: Section IV-D deploys the twin as a feedforward dynamics model on two physical robots, and Section IV-E exports it as a simulation asset for policy learning. All experiments follow the capture protocol of Section IV-A.

A. Capture Protocol and Implementation

We capture all interactions with the handheld Hoi! gripper [10], which provides posed RGB-D observations, the tool trajectory from Project Aria MPS [32], and the contact wrench from a wrist-mounted force-torque sensor. Once a part is grasped, we open and close it over a range of speeds and accelerations chosen to excite the individual regressor axes: slow, low-acceleration motion for Coulomb friction, a wide speed spectrum for viscous damping, high accelerations for inertia, and passive following for the mechanism. Fig. 4(b) shows an example probing sequence. In our implementation, FunFact [18] reconstructs panels, handles and functional relations from the same recording and therefore already shares the probing frame; we identify the manipulated handle by proximity to the tracked TCP, use FunFact’s handle-to-panel relation to select the associated door or drawer panel, attach the recovered model to that pair, process the instance meshes, and export the articulated asset to USD. Because $\mathcal{M}_{\text{phys}}$ is independent of the geometric representation, it can equally be registered onto an existing asset: the real-to-sim free-swing experiments of Section IV-B use the exported asset, while the policy-learning experiments of Section IV-E attach the identified dynamics and mechanism to the door asset of the training pipeline, leaving its geometry unchanged. In both cases the identified $\hat{\beta}$ is written directly to the joint’s inertia, friction and damping fields, so the parametric half of the twin is native to the

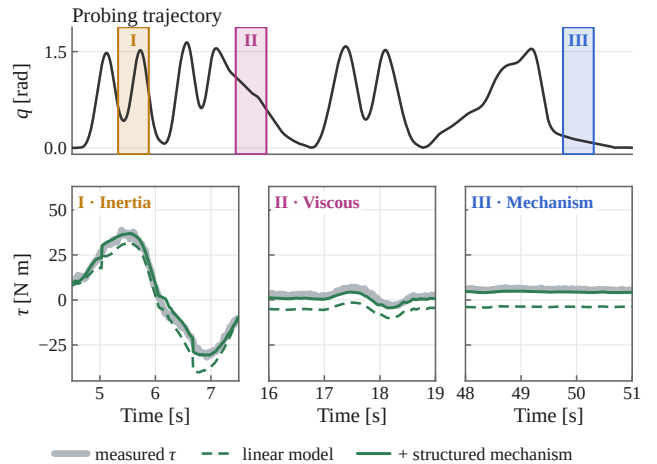


Fig. 4: **Dynamics model fit for the wood door.** (a) Measured generalized effort (gray) against the parametric model alone (dashed) and with the structured mechanism added (green); (b) the probing trajectory. The parametric model tracks the shape of the effort but is offset from it by a configuration-dependent amount, which the learned mechanism recovers. Shaded bands mark the probing regimes for inertia (yellow), viscous damping (purple), and the mechanism (blue).

simulator, while the mechanism term is evaluated per step as a state-dependent joint effort. An independently obtained representation has to be registered to the probing frame first.

B. Estimation Fidelity

We evaluate estimation accuracy at three levels: kinematic parameters, compared against annotated ground-truth axes on the real-world test rig; inertial parameters, whose ground truth can be independently measured (unlike friction parameters, for which no such reference exists); and finally aggregate system fidelity, where kinematic and dynamic terms are evaluated jointly.

Kinematics Estimation. We evaluate articulation types and axes on our real-world test bench against ground truth annotated in a Leica RTC360 laser scan [10], [17]. After registering estimates to the scan, we measure axis-angle errors for both joint types and axis-position errors for revolute joints [2], [17]. We compare against MoMa-SG [2], which tracks object points during non-instrumented human interaction and is run on our interactions, and OPD-Multi [14], which estimates articulation from visual input without interaction. Interaction-based methods outperform the visual baseline (Table I). OPD-Multi estimates prismatic directions reasonably well (4.8°), but yields large revolute-axis errors (34.9°) and only 54.1% revolute type accuracy; both interaction-based methods classify all joints correctly. Compared with MoMa-SG, our method slightly improves revolute-axis orientation (1.5° vs. 1.7°) and halves position error (23 vs. 46 mm). We attribute this improvement to directly measured tool poses, avoiding reliance on surface-point tracking susceptible to low texture and motion blur.

Dynamics and Mechanism Estimation. We next assess the fidelity of our estimated dynamics parameters against a prior-based paradigm such as [5]. We focus on the inertial parameters, since ground truth is readily obtainable for them.

TABLE I: **ARTICULATION AXIS ESTIMATION.** Evaluated on the real-world test bench against 12 axes annotated in a Leica RTC360 laser scan. Metrics follow ArtiPoint [17]: θ_{err} is the angle between axis lines, folded to $[0, 90]^\circ$, and d_{L_2} the distance between revolute axis lines along their common perpendicular. Errors accumulate only over predictions with the correct joint type. Values are means with the sample standard deviation in smaller type; † no deviation available. **Bold** indicates best performance, underline second best; tied entries are all marked best.

Method	Prismatic		Revolute		Type acc. [%] $\uparrow$	
	$\theta_{\text{err}} [^\circ] \downarrow$	$d_{L_2} [\text{mm}] \downarrow$	$\theta_{\text{err}} [^\circ] \downarrow$	$d_{L_2} [\text{mm}] \downarrow$	Pris.	Rev.
FORCETWIN (Ours)	1.4 ± 0.2	1.5 ± 0.8	23 ± 26	100.0	100.0	100.0
MoMa-SG [2]	<u>1.6 ± 0.8</u>	<u>1.7 ± 0.0</u>	<u>46 ± 20</u>	100.0	100.0	100.0
OPDMulti [14]	$4.8^\dagger$	34.9 ± 29.1	143 ± 117	100.0	100.0	54.1

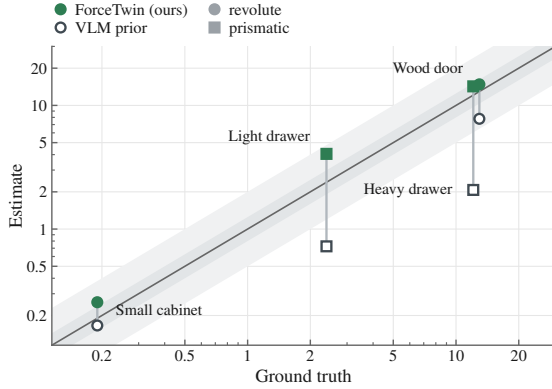


Fig. 5: **Inertial-parameter fidelity on the four objects for which ground truth is available:** estimate against ground truth on logarithmic axes, so an estimate on the diagonal is exact and vertical distance from it is the error. Marker shape distinguishes revolute (circle) from prismatic (square) joints, whose parameters carry the two units given on the axes. Nested shaded bands mark agreement to within $\pm 25\%$ (inner) and to within a factor of two (outer). The macro mean of the absolute relative errors is 29.5% for FORCETWIN and 53.0% for the VLM prior.

For prismatic joints this is the mass, measured with a scale. For revolute panels, the hinge inertia follows from mass and geometry as $I = \frac{1}{3}m\ell^2$ under the assumption of constant density, while compact parts such as locks and handles are treated as point masses with $I = m\ell^2$. Here ℓ is the distance from the hinge axis: the panel’s extent from the hinge in the first case, and the radial offset of the part in the second. No ground truth is available for the parametric friction parameters or for the nonlinear MLP. To still evaluate these terms, we assess the aggregate fit at the system level: the real-to-sim free-swing experiment below compares the closing behavior of the real object with that of its simulated twin. Fig. 4 shows a qualitative example of the fit. The parametric model tracks the measured effort qualitatively, but with a non-constant offset that reveals the mechanism it cannot represent. Aggregated over all objects (Fig. 5), the interaction-based estimate is more accurate than the prior-based one (29.5% vs. 53.0% macro mean absolute relative error), and it is the better estimate on three of the four objects for which ground truth is available. Our accuracy improves systematically with the magnitude of the inertial parameter: the two heaviest parts are estimated to within

10.4% and 13.7%, whereas the lightest ones reach 63.8% and 30.2%. We attribute this to the signal-to-noise ratio of the measurement rather than to the estimator itself, as light parts excite only small interaction forces, so that the force-torque noise band occupies a larger fraction of the measured signal. Consistently, the sole case in which the prior-based baseline is more accurate is the smallest object in the set. The prior-based baseline underestimates the inertial parameter for every object, despite being provided with scale information and prompted to reason about material. The largest error occurs for the heavy drawer (83.5%), whose effective mass is dominated by its contents, hidden state that cannot be inferred from appearance. Our estimator, in turn, overestimates on all four objects. The bias is consistent in sign, which suggests that residual unmodeled effort is partially absorbed into the inertial term during the parametric fit. This reflects a structural property of the decomposition rather than insufficient optimization: both the parametric terms and the mechanism response depend on the articulation velocity, so the split between them is only partially identifiable from torque observations, and the fit objective is predictive accuracy rather than parameter recovery. The downstream uses of the twin consume the full torque model, whose aggregate fidelity is assessed at system level by the free-swing experiment below.

System-Level Fidelity: Real-to-Sim Free Swing. The preceding experiment evaluates individual parameters, but only the inertial ones admit ground truth. We therefore assess the twin at the system level, where inertia, friction and mechanism act jointly. We release the metal door and the wood door from 45° and 90° and compare the measured closing time against simulations parameterized either by our interaction-derived estimates or by the VLM-prior baseline. Our estimates yield the lower error in all four conditions (Table III), reducing the macro mean from 57.7% to 16.3%. The VLM-prior baseline closes every door in roughly 2 s regardless of door or release angle, indicating that it does not match the damped dynamics closely. Note that this measure is aggregate and not injective: a large mismatch proves the dynamics are wrong, but a close match does not uniquely identify the individual parameters, since compensating errors could in principle yield the same closing time. It therefore complements the parameter-wise evaluation above.

C. Model-Class Ablations

We evaluate the benefit of added model complexity. Adding viscous damping to the Coulomb-inertia model reduces in-sample effort RMSE by up to 12%. Adding the structured mechanism reduces it by a further 67–73% on mechanism-dominated parts such as the doors and the oven, against 12% on the sliding door, which has no mechanism. We fit the two stages sequentially, the parametric coefficients first and the mechanism on their residual. Alternating between them changes the fit RMSE by less than 5% of the parametric baseline while redistributing effort between overlapping terms, so we do not iterate.

TABLE II: **REAL-ROBOT ARTICULATION.** Spot and Franka FR3 track robot-specific opening goals within 6s using a fixed controller. FORCETWIN uses interaction-identified dynamics; VLM prior uses visual and language estimates [5], [6]; Kinematics only omits dynamics compensation [2], [3]. *Complete*: maximum goal completion (capped at 100%). *Track*: goal-normalized tracking RMSE (%). Values are means with sample SD; the overall mean equally weights nine object-robot pairs. *Mechanism* rates the contribution of built-in mechanisms to articulation force. Objects are ordered by FORCETWIN tracking error within each robot panel. **Bold/underline**: best/second best per object and metric, with ties marked best.

	Object	Mechanism	FORCETWIN (Ours)		VLM prior		Kinematics only	
			Complete $\uparrow$	Track $\downarrow$	Complete $\uparrow$	Track $\downarrow$	Complete $\uparrow$	Track $\downarrow$
Spot	Drawer	Weak	88.0 $\pm$ 1.8	9.2 $\pm$ 2.9	95.2 $\pm$ 4.0	<u>34.1</u> $\pm$ 3.1	<u>90.6</u> $\pm$ 3.7	38.0 $\pm$ 2.6
	Sliding door	None	100.0 $\pm$ 0.0	10.5 $\pm$ 1.9	100.0 $\pm$ 0.0	<u>15.4</u> $\pm$ 1.9	100.0 $\pm$ 0.0	18.5 $\pm$ 2.6
	Wood door	Medium	100.0 $\pm$ 0.0	14.0 $\pm$ 1.5	<u>91.2</u> $\pm$ 0.9	<u>24.3</u> $\pm$ 0.0	88.3 $\pm$ 4.7	27.3 $\pm$ 1.6
	Oven	Strong	100.0 $\pm$ 0.0	33.3 $\pm$ 3.1	<u>4.0</u> $\pm$ 0.7	<u>91.1</u> $\pm$ 0.4	2.2 $\pm$ 0.1	91.9 $\pm$ 0.2
	Metal door	Strong	N/A (object forces exceed Spot’s actuation limits)					
Franka	Sliding door	None	93.7 $\pm$ 0.7	10.5 $\pm$ 0.4	<u>98.7</u> $\pm$ 1.0	<u>17.2</u> $\pm$ 0.9	99.3 $\pm$ 1.0	26.9 $\pm$ 12.2
	Drawer	Weak	<u>85.6</u> $\pm$ 9.6	17.6 $\pm$ 3.2	93.1 $\pm$ 3.2	37.0 $\pm$ 0.4	85.1 $\pm$ 11.2	<u>22.4</u> $\pm$ 4.2
	Metal door	Strong	81.2 $\pm$ 3.5	24.4 $\pm$ 5.5	<u>1.6</u> $\pm$ 0.0	<u>91.0</u> $\pm$ 0.0	1.5 $\pm$ 0.0	<u>91.0</u> $\pm$ 0.0
	Wood door	Medium	69.8 $\pm$ 19.4	30.3 $\pm$ 11.0	<u>51.7</u> $\pm$ 4.0	<u>47.7</u> $\pm$ 2.8	44.0 $\pm$ 21.3	56.0 $\pm$ 16.6
	Oven	Strong	67.6 $\pm$ 0.0	49.7 $\pm$ 0.0	<u>1.8</u> $\pm$ 1.0	<u>90.8</u> $\pm$ 0.9	0.9 $\pm$ 0.1	91.6 $\pm$ 0.1
Overall mean			87.3	22.2	<u>59.7</u>	<u>49.8</u>	56.9	51.5

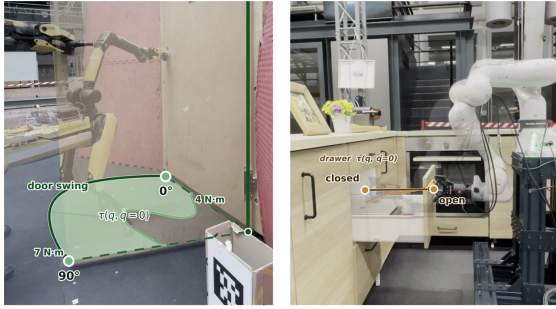


Fig. 6: **Real-world experiments of zero-shot model-based articulation** (Table II) across two embodiments: Spot and Franka FR3. We visualize the initial and final articulation states as blended overlays for both embodiments. The recovered quasi-static mechanism torque for each object is plotted over the opening trajectory, highlighting the variation in mechanism behavior across objects.

D. Does Dynamics Fidelity Change Manipulation?

This experiment evaluates whether the fidelity of a twin’s dynamics influences manipulation performance: We use ForceTwin’s object dynamics as a feedforward term of a Cartesian impedance controller,

$$\mathbf{f}_{\text{cmd}} = K_x(x_d - x) + D_x(\dot{x}_d - \dot{x}) + \mathbf{f}_{\text{ff}},$$

$$\mathbf{f}_{\text{ff}} = \left(J_p(q)^\top \right)^\dagger \left[\hat{I} \ddot{q}_d + \hat{\mu} \operatorname{sgn}(\dot{q}_d) + \hat{b} \dot{q}_d + \hat{\tau}_{\text{mech}}(q_d, \dot{q}_d) \right]. \quad (11)$$

where the bracket sums the generalized effort of the identified inertia, Coulomb and viscous damping, and mechanism response along the reference articulation trajectory $q_d(t)$, and the transposed pseudoinverse of the object Jacobian’s linear block J_p maps them to an end-effector force added to the impedance feedback. The twin enters only through $\mathbf{f}_{\text{ff}}$, so we vary its source across the two prevalent twin-construction paradigms and ours, keeping robot, controller, and task fixed: kinematics-only twins [2], [3], [14], [33] provide J_p but no dynamics ($\mathbf{f}_{\text{ff}} = \mathbf{0}$), VLM-prior twins [5], [6] supply prior-based parameter estimates to the same feedforward, and FORCETWIN supplies the parameters identified from instrumented interaction. In each trial, the robot starts with the handle grasped and tracks $q_d(t)$ from the closed configuration to an embodiment-specific goal within 6s (Fig. 6);

TABLE III: **REAL-TO-SIM FREE-SWING FIDELITY.** Closing times of the physical door and of twins parameterized by our interaction-derived estimates and by the VLM prior, released from two initial opening angles q_0 . Times are in seconds and the absolute relative error (ARE) is a percentage. The final row is the equal-weight macro mean. Only two estimates are compared, so **bold** marks the lower ARE and no second rank is reported.

Object	q_0 [°]	ForceTwin		VLM prior	
		Real [s]	t [s]	ARE $\downarrow$ [s]	ARE $\downarrow$
Metal door	45	5.6	5.9	5.4	2.1
Metal door	90	8.4	8.8	4.8	2.5
Wood door	45	3.7	5.0	35.1	2.0
Wood door	90	5.0	6.0	20.0	2.4
Macro mean	—	—	—	16.3	—

We report goal completion and goal-normalized tracking RMSE. Table II reports goal completion and tracking error on five objects across Spot and Franka FR3 under a common 6s goal. FORCETWIN yields the lowest tracking error on every object and the highest macro mean completion (87.3%, against 59.7% for the VLM prior and 56.9% for kinematics only), and the margin scales with the dynamic load. On the drawers and sliding doors all three models complete the goal, and the identified model improves tracking error by factors of 1.3 to 4. On the strong-mechanism objects, the oven on both robots and the metal door on the Franka, both baselines stall below 5% completion because their feedforward supplies only a fraction of the required force, while the identified model reaches 67–100%. The wood door lies between these regimes on both embodiments, where the baselines lose 9–26 percentage points and roughly double the tracking error.

E. The Twin as a Training Asset

Beyond the feedforward evaluations above, we show that the identified models can be used directly for reinforcement learning. We register the identified wood- and metal-door dynamics, including the recovered nonlinear closing mechanism, onto the door asset in Isaac Lab [34]. Using the motion-imitation pipeline from [35], we train whole-body door-traversal policies. That pipeline randomizes the door parameters during training; here we replace that randomization with the identified dynamics, leaving the policy architecture and training procedure unchanged. We deploy both policies on an ANYmal quadruped with an arm and evaluate them on the two real doors. The wood-door policy succeeds on the wood door in all five trials. The metal-door policy also succeeds on this door, but exhibits a stop-and-go motion: it advances the door and then pauses while the imitation phase catches up (Fig. 7). On the metal door, the metal-door policy succeeds, whereas the wood-door policy fails its initial opening attempt. These experiments demonstrate that the identified models integrate into an existing policy-learning pipeline and support real-world deployment on contact-rich whole-body tasks. Whether training with identified dynamics

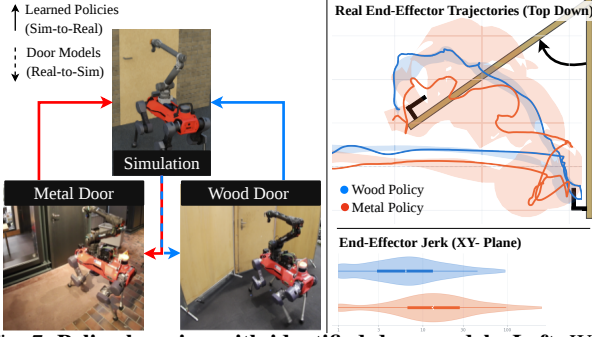


Fig. 7: **Policy learning with identified door models.** **Left:** We use the identified wood- and metal-door models (real-to-sim) to train whole-body policies, which we then deploy on ANYmal (sim-to-real). **Right:** Both policies on the real wood door; shaded regions show the average over five trials. The wood-door policy moves the end-effector smoothly, while the metal-door policy stops and starts, pulling hard and then pausing, which gives it higher jerk (bottom) due to the worse sim-to-real calibration.

consistently improves performance over domain randomization requires a controlled comparison, which we leave to future work.

V. LIMITATIONS

Our decomposition of the effort model is not unique: a constant mechanism damping is absorbed by b , and any configuration-dependent load by g_θ , including gravity when the joint axis is not vertical. The model is memoryless in $(q, \dot{q})$, so hysteresis, backlash, stiction at rest, and latch states lie outside its class, and both g_θ and the joint range hold only where the object was probed. Further, identification is per instance and needs physical interaction: each part is probed individually, and the probing has to excite the individual regressor axes, so passive demonstration recordings do not suffice. Priors scale across a scene where probing does not, so the two are complementary, and transferring identified dynamics to parts that were not probed is left to future work. Identification is also one-shot: a loaded drawer or a readjusted closer changes the object, and refining a twin online from the robot’s own interaction forces remains open.

VI. CONCLUSION

FORCETWIN identifies physics-informed digital twins of articulated objects from instrumented human interaction. A person probes the object with a handheld force-sensing gripper, and from the tool trajectory and contact wrenches we recover the articulation, effective inertia, Coulomb and viscous terms under nonnegativity constraints, and a residual for the nonlinear mechanism. The identified parameters nearly halve the inertial error of a prior-based estimate and reproduce measured door-closing times. A single twin then serves two uses: as the feedforward model of an impedance controller, it operates objects on which kinematics-only and prior-based twins stall, and exported to simulation it is used to train whole-body door-traversal policies transferring to the real objects.

ACKNOWLEDGEMENTS

This work was supported by SNSF Advanced Grant 216260, the Lamarr Institute, and Google. The authors used generative AI to assist with drafting and editing the manuscript, cleaning up figures and coding; all content was reviewed by the authors, who take full responsibility for it.

REFERENCES

- [1] Q. Gu *et al.*, “ArtiSG: Functional 3D Scene Graph Construction via Human-demonstrated Articulated Objects Manipulation,” *arXiv:2512.24845*, 2026.
- [2] M. Büchner *et al.*, “Articulated 3D Scene Graphs for Open-World Mobile Manipulation,” *arXiv:2602.16356*, 2026.
- [3] Z. Huang *et al.*, “REACT3D: Recovering Articulations for Interactive Physical 3D Scenes,” *IEEE RA-L*, 2026.
- [4] A. Yu *et al.*, “Pandora: Articulated 3D Scene Graphs from Egocentric Vision,” *arXiv:2603.28732*, 2026.
- [5] D. Iliash *et al.*, “Artiverse: A Diverse and Physically Grounded Dataset for Articulated Objects,” in *CVPR*, 2026.
- [6] H. Li *et al.*, “PhysGraph: A Physics-aware 3D Scene Graph for Perception and Reasoning,” *arXiv:2606.08655*, 2026.
- [7] D. Katz and O. Brock, “Manipulating Articulated Objects with Interactive Perception,” in *ICRA*, 2008.
- [8] R. Martín-Martín and O. Brock, “Coupled Recursive Estimation for Online Interactive Perception of Articulated Objects,” *IJRR*, 2022.
- [9] R. Buchanan *et al.*, “Online Estimation and Manipulation of Articulated Objects,” *Auton. Robots*, 2026.
- [10] T. Engelbracht *et al.*, “Hoi! - A Multimodal Dataset for Force-Grounded, Cross-View Articulated Manipulation,” in *CVPR*, 2026.
- [11] H. Choi *et al.*, “In-the-Wild Compliant Manipulation with UMI-FT,” *arXiv:2601.09988*, 2026.
- [12] J. Bohg *et al.*, “Interactive Perception: Leveraging Action in Perception and Perception in Action,” *IEEE T-RO*, 2017.
- [13] J. Sturm *et al.*, “A Probabilistic Framework for Learning Kinematic Models of Articulated Objects,” *JAIR*, 2011.
- [14] X. Sun *et al.*, “OPDMulti: Openable Part Detection for Multiple Objects,” *arXiv:2303.14087*, 2023.
- [15] J. Liu *et al.*, “PARIS: Part-level Reconstruction and Motion Analysis for Articulated Objects,” *arXiv:2308.07391*, 2023.
- [16] A. Jain *et al.*, “ScrewNet: Category-Independent Articulation Model Estimation From Depth Images Using Screw Theory,” *arXiv:2008.10518*, 2021.
- [17] A. Werby *et al.*, “Articulated Object Estimation in the Wild,” *arXiv:2509.01708*, 2025.
- [18] Z. Fu *et al.*, “FunFact: Building Probabilistic Functional 3D Scene Graphs via Factor-Graph Reasoning,” *arXiv:2604.03696*, 2026.
- [19] R. Buchanan *et al.*, “Online Estimation of Articulated Objects with Factor Graphs using Vision and Proprioceptive Sensing,” *arXiv:2309.16343*, 2023.
- [20] F. Endres *et al.*, “Learning the Dynamics of Doors for Robotic Manipulation,” in *IROS*, 2013.
- [21] A. Jain *et al.*, “Improving Robot Manipulation with Data-Driven Object-Centric Models of Everyday Forces,” *Auton. Robots*, 2013.
- [22] Z. Jiang *et al.*, “Ditto: Building Digital Twins of Articulated Objects from Interaction,” in *CVPR*, 2022.
- [23] C. Chi *et al.*, “Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots,” *arXiv:2402.10329*, 2024.
- [24] Q. Chen *et al.*, “URDFormer: A Pipeline for Constructing Articulated Simulation Environments from Real-World Images,” in *RSS*, 2024.
- [25] L. Ma *et al.*, “Sim2Real²: Actively Building Explicit Physics Model for Precise Articulated Object Manipulation,” in *ICRA*, 2023.
- [26] M. Villasevil *et al.*, “Reconciling Reality through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation,” in *RSS*, 2024.
- [27] T. Dai *et al.*, “Automated Creation of Digital Cousins for Robust Policy Learning,” *arXiv:2410.07408*, 2024.
- [28] H. Jiang *et al.*, “PhysTwin: Physics-Informed Reconstruction and Simulation of Deformable Objects from Videos,” in *ICCV*, 2025.
- [29] Y. Chebotar *et al.*, “Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience,” in *ICRA*, 2019.
- [30] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [31] G. Schwarz, “Estimating the Dimension of a Model,” *The Annals of Statistics*, 1978.
- [32] J. Engel *et al.*, “Project Aria: A New Tool for Egocentric Multi-Modal AI Research,” *arXiv:2308.13561*, 2023.
- [33] A. Delitzas *et al.*, “FunRec: Reconstructing Functional 3D Scenes from Egocentric Interaction Videos,” in *CVPR*, 2026.
- [34] M. Mittal *et al.*, “Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning,” *arXiv:2511.04831*, 2025.
- [35] J.-P. Sleiman *et al.*, “Guided Reinforcement Learning for Robust Multi-Contact Loco-Manipulation,” *arXiv:2410.13817*, 2024.