# What Explicit Physics Buys a Manipulation World Model

Di Yang[1,†] Muyang Liu[1] Junzhou Chen[1] Yufei Wang[2,*] Yanhai Xiong[1,*]
[1]College of William & Mary [2]SparcAI Inc.
dyang06@wm.edu mliu22@wm.edu jchen57@wm.edu yxiong05@wm.edu
yufei.wang@sparclab.ai

## Abstract

*The outcomes of manipulation actions often depend on physical properties that are not directly observable from a single image. A drawer may be locked despite appearing movable, and a tall object may slide or topple depending on its friction and mass distribution. We ask when explicit information about such a property helps a world model that already observes interaction history. Across six simulated manipulation scenes, we compare models given the current observation, observation history, or history with an estimated or an oracle physical factor, under matched training. In scenes where history leaves the factor ambiguous, an estimated mobility factor improves the standard motion prediction error by only 4%, yet it reduces pushing against immovable targets by 37%. Most of this stopping improvement comes from constraining the predicted motion with the factor. Applying the constraint to a history model at test time recovers 87% of the benefit without retraining. Conditioning the learned dynamics on the factor gives smaller, scene-dependent gains on movable objects and reduces toppling when pushing tall objects. A simple contact-based stopping rule still stops better than every learned setting without oracle information. On a real arm, the test-time constraint and the contact rule also shorten the push against immovable targets from 195 to 152 and 40 mm. These results separate what a world model gains by learning from physical information from what direct constraints and contact feedback already provide.*

## 1. Introduction

World models predict how a scene changes under a robot's actions and let a planner choose actions by imagining their outcomes [2, 11, 12, 14, 66]. For manipulation, the outcome of a push often depends on physical properties that are absent from the image. Two cabinet doors can look identical while one of them is locked. A bottle pushed near its top can slide or tip over, depending on its friction and the height of its center of mass. A model that sees only the current observation must guess such a property from appearance. A model that also sees recent interaction can infer it from how the scene has responded so far.

A natural alternative is to estimate the property with a dedicated module and give it to the world model explicitly. Explicit physical parameters improve adaptation and prediction in several learned systems [21, 24, 64]. In world models, the value of such information is usually judged by prediction error. In our experiments an estimated mobility factor lowers the moving-point error of a strong 3D world model by only 4% over a model that already sees observation history.

Standard prediction metrics need not capture the errors that matter for planning [23, 45, 62]. The moving-point error is measured on targets that are free to move. A planner facing a locked target, however, mainly needs to know whether the target will move at all. A small difference in prediction accuracy can therefore accompany a large difference in closed-loop behavior. We measure the value of an explicit factor by the decisions it changes, and we ask which part of that value requires the factor to be inside the world model.

We build six scenes in ManiSkill [41] (Fig. 1). In four of them the robot pushes a part of an articulated object from PartNet-Mobility [54], which may be free, locked, or stopped by a hidden limit. In the fifth it pushes one object in a tabletop clutter where some objects are fixed to the table. In the sixth it pushes a tall object whose friction and mass distribution decide whether it slides, rocks or topples. For each scene we train PointWorld [14], a 3D point-flow world model, in four settings: the current observation alone, five frames of observation history, and history with an estimated or an oracle factor. All settings share data, architecture and optimization, and each is trained with three seeds. We evaluate every model on held-out objects by single-step prediction and by model-predictive control. DINO-WM [66], which predicts in the feature space of a pretrained image encoder, serves as a second model family. On a real SO-101 arm, we repeat the stopping comparison with PointWorld models fine-tuned on real pushes (Sec. 4.8).

*Corresponding author. †Project lead. *Preprint.*

In closed loop the estimated factor shortens the push wasted on locked targets by 37% where history leaves the factor ambiguous. The factor reaches PointWorld in two ways, as an input to the learned dynamics and as a projection of the predicted motion onto the motions it allows. Controls show that the projection carries most of the stopping behavior, while conditioning the dynamics on the factor mainly affects how well movable objects are pushed. A simple rule on contact force and target motion stops earlier than every learned setting without oracle information.

Our contributions are threefold. First, we compare implicit, history-based and explicit physical information under matched budgets, across six scenes and two model families, with closed-loop evaluation on a locked test split. Second, in a point-flow world model, we separate the value of an explicit mobility factor into a stopping effect carried by an output constraint and a smaller effect of factor conditioning on movable objects. We set both against a larger model and a model-free stopping rule. Third, we show that phantom target motion captures a decision-relevant error: the estimated factor reduces it by 40% and the wasted push by 37%, but the moving-point error by 4%.

## 2. Related Work

**World models for manipulation.** Action-conditioned world models have been learned in latent spaces trained by reconstruction [11, 12] or by latent consistency [13], in pixel space with diffusion [1], in object-centric latents [8], and in the feature spaces of pretrained visual encoders [2, 42, 66]. Particle and graph models predict the motion of points or object parts and plan through it [26, 37, 38, 64], and recent 3D point world models scale this idea to diverse objects and real scenes [6, 14, 57]. PointWorld [14] predicts the 3D motion of scene points from RGB-D input with gripper points as the action and plans with model-predictive path integral control [52]. For articulated objects, Local World Models [30] learn joint-type-agnostic part dynamics from part point clouds, affordance points and proprioception and study their scaling. These models receive physical properties only through what the observation and the interaction reveal. We use two of them, PointWorld and DINO-WM, to ask what changes when a hidden property is supplied explicitly.

**Explicit and implicit physical information.** One line of work supplies physical parameters to learned dynamics, as node features of learned simulators [37, 64], as modulation of the network [24], or as morphology descriptors [5, 56]. Others identify differentiable physics from images [25, 46], and a structured use of a parameter can make predictions follow counterfactual changes of it [3, 50]. Another line infers the parameters, or a belief over them, from interaction history [21, 22, 58]; in legged locomotion such a latent outperforms explicitly identified parameters while an oracle remains best [21]. The properties themselves can be estimated from interaction [9, 15, 16, 55], vision-language priors [47] or video models [60]. Articulation models are recovered from interaction [18, 27, 33], tracking [49] or video [61]. They guide manipulation through actionability maps [29] and articulation flow [7], or replace a learned model by an explicit digital twin [17]. Closest to our question, Tan *et al*. [40] measure which physical parameters latent predictive representations retain, and Joseph *et al*. [19] probe how video world models encode physical quantities. Thorpe *et al*. [43] argue that world models must represent hidden physical structure explicitly. None of these works compares implicit, history-based and explicit physical information within one learner under matched budgets in closed-loop control.

**Evaluating world models by the decisions they support.** Model accuracy and control performance can disagree [23, 51], which has motivated models trained to be accurate where the decision requires it [10, 44] and analyses of what a latent state must retain for control [20]. Closed-loop and physics benchmarks find that visual quality does not guarantee task success or physical faithfulness [4, 62, 63]. Yu *et al*. [59] argue for decision-centered evaluation of world models. Wang *et al*. [45] show in controlled planning domains that the decision-relevant part of the prediction error predicts planning quality where the total error does not. Our phantom motion is an instance of this idea for a target that cannot move.

## 3. Study Design

Our goal is to measure how the value of a physical factor changes with the way the information reaches the world model, while everything else stays fixed. Figure 2 shows the three ways in which physical information can reach a decision in our study.

### 3.1. Problem formulation

At time $t$ the robot observes the scene with two calibrated RGB-D cameras. Following PointWorld [14], the observation is a set of scene points $X_t = \{x_t^i\}_{i=1}^N$, $x_t^i \in \mathbb{R}^3$, with appearance features, and the action is represented by points sampled on the parallel-jaw gripper. The robot commands gripper increments $\delta_\tau \in \mathbb{R}^3$, and $G_{t:t+T}$ denotes the gripper points along the commanded trajectory over the next $T = 10$ control steps. We write the scene evolution in simplified form as $X_{t+1} = f(X_t, \delta_t; z)$, where $z$ collects hidden physical properties that are not a function of the current observation. A world model $F_\theta$ predicts the displacements of all scene points over the horizon,

$$\hat{\Delta}_{t+1:t+T} = F_\theta\big(X_t,\ G_{t:t+T},\ c_t\big), \qquad \hat{x}^i_{t+\tau} = x^i_t + \hat{\Delta}^i_{t+\tau}, \tag{1}$$

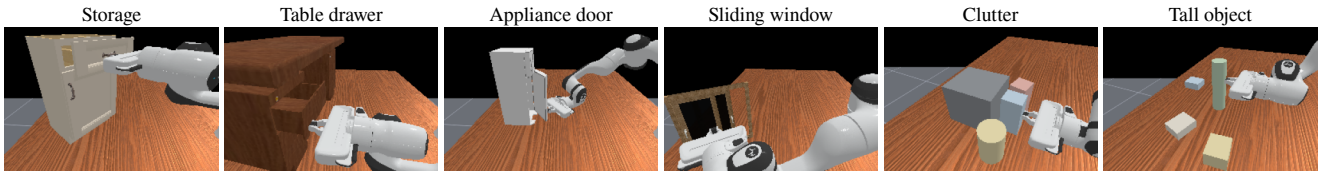



Figure 1. The six scenes. In the first four, the pushed part of an articulated object is free, locked, or stopped by a hidden limit. In the clutter scene, the target and the object in its path are each either movable or fixed to the table. In the last scene, the friction and the height of the center of mass of a tall object decide whether a push at height makes it slide, rock or topple. In every scene the hidden property cannot be read from a single image.

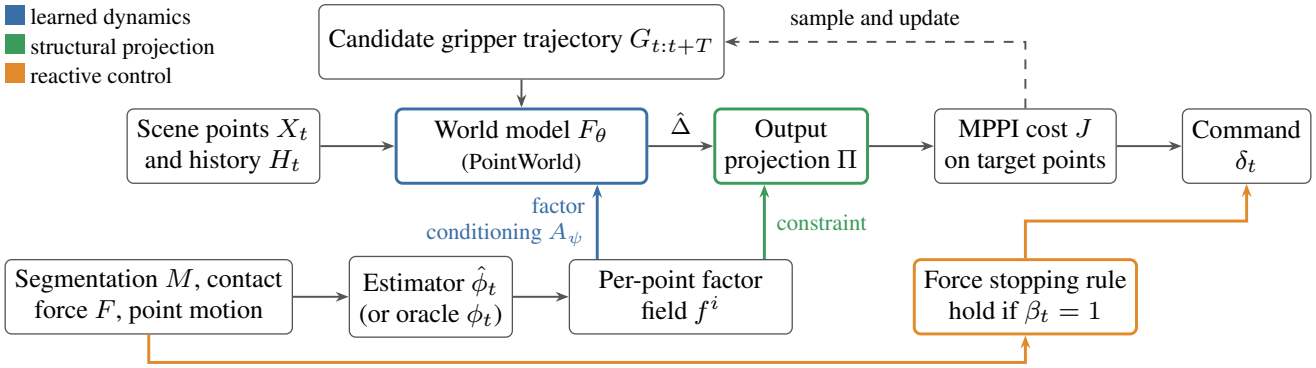



Figure 2. How physical information reaches the decision. The world model predicts the motion of the scene points for candidate gripper trajectories, and MPPI scores the predicted motion of the target. An explicit factor, estimated from segmentation, contact force and point motion or given by an oracle, can condition the learned dynamics (blue) or constrain the predicted motion through an output projection (green). A force stopping rule acts on the same sensing without a world model (orange).

where $c_t$ is the additional information that distinguishes the four settings of Sec. 3.3. Within PointWorld, all four information settings are trained with the same objective,

$$\mathcal{L}(\theta) = \frac{1}{T\,|V|} \sum_{\tau=1}^{T} \sum_{i \in V} \left\| \mathcal{N}_\tau\big(\hat{\Delta}^i_{t+\tau}\big) - \mathcal{N}_\tau\big(\Delta^i_{t+\tau}\big) \right\|_2^2, \quad (2)$$

where $\Delta^i_{t+\tau} = x^i_{t+\tau} - x^i_t$ is the true displacement, $V$ the set of visible points, and $\mathcal{N}_\tau$ standardizes displacements with statistics computed once per scene on the training split. This is the released PointWorld objective without its learned uncertainty term.

### 3.2. Scenes and hidden factors

**Mobility.** The first factor is the mobility of the pushed body, that is, whether it can move and how far. Four scenes use articulated objects from PartNet-Mobility [54] simulated in ManiSkill [41], namely storage furniture with drawers and revolute doors, table drawers, appliance doors and sliding windows. For a part with joint coordinate $q \in [q_{\min}, q_{\max}]$, the hidden state restricts the reachable interval to $[q_t - u_t^-,\ q_t + u_t^+]$ around the current value $q_t$, where $u_t^\pm$ is the remaining travel in each direction. The part is free (travel given by the joint limits), locked ($u_t^+ = u_t^- = 0$), or limited. A limited part has a stop between 5 and 30 mm (prismatic) or 0.05 and 0.30 rad (revolute) from its joint value at the start of the episode. Its appearance is identical in all three states. The clutter scene places four to six boxes and cylinders on a table, with one neighbor always in the path of the target. Objects belong to a light-colored item class that is movable with probability 0.8 or to a dark fixture class that is fixed with probability 0.8. Appearance is thus informative, but a hidden minority contradicts it. An episode is free when the target and its path neighbor are both movable, locked when the target is fixed, and limited when the target moves until it meets a fixed neighbor.

**Tipping.** The second factor governs a dynamic outcome rather than a kinematic constraint. A tall box or cylinder of height $h \in [12, 24]$ cm and half width $r$ is pushed at height $h_p \in [0.45h, 0.85h]$. Its friction coefficient $\mu$ is drawn from $[0.10, 0.55]$ and the height of its center of mass from $[0.25h, 0.55h]$. A quasi-static push starts a tilt rather than a slide when $\mu\, h_p > r$, and the height of the center of mass then decides whether the object rocks back or topples. We call an object tippable when $\mu > r/h_p$.

**Data.** Training clips come from scripted pushes against the visible face of the target, mixed with slow pushes, pauses, retreats and side steps, so that the models see parts both move and stop. A clip holds the current frame, five history frames and ten future frames. Training and evaluation use disjoint objects or layouts, with 13.7k to 19.6k training clips per scene (Sec. A).

### 3.3. Information settings

The four settings differ in the physical information supplied to the model through $c_t$ of Eq. (1),

$$c_t^{\mathrm{I}} = \varnothing, \quad c_t^{\mathrm{H}} = H_t, \quad c_t^{\mathrm{E}} = \big(H_t,\, \hat{\phi}_t\big), \quad c_t^{\mathrm{O}} = \big(H_t,\, \phi_t\big), \tag{3}$$

for Implicit, History, History + estimated factor and History + oracle factor. In PointWorld, the two factor settings additionally constrain the predicted motion through the output projection of Sec. 3.4. The history $H_t = \{x_{t-k}^i - x_t^i\}_{k=1}^5$ holds the displacement of every scene point over the five previous frames. It carries the evidence of the warm-up interaction.

The factor $\phi_t = \phi(z, X_t)$ is a per-point field $f^i \in \mathbb{R}^{16}$; it depends on the state because the remaining travel does. For mobility, a point of the pushed part with joint axis $a$ and pivot $o$ receives

$$f^i = \big(a,\; x_t^i - o,\; d^i,\; \omega,\; \gamma,\; m^i,\; u_t^+,\; u_t^-\big), \tag{4}$$

where $d^i$ is the unit direction of allowed motion, $\omega$ the angular axis ($a$ for a revolute and zero for a prismatic joint), $\gamma$ a confidence and $m^i$ the part membership; channels are zero outside the part. In the clutter scene the geometry channels are empty and $u_t^{\pm}$ is zero for the points of fixed objects. For tipping the field holds the offset of each point of the target from its center of mass and the friction coefficient $\mu$.

The estimate $\hat{\phi}_t$ uses the kinds of signals that a deployed robot can measure; in our simulation the point tracks come from the simulator. It assumes a part or instance segmentation $M$, which we take from the simulator and which therefore acts as a perfect segmenter. It also uses the contact force $F$ between gripper and target, which a robot can derive from its joint torques. When at least 16 part points have moved by more than 2 mm, a rigid fit

$$(R^\star, b^\star) = \arg\min_{R \in SO(3),\, b} \sum_{i \in M} \big\| R\, x_{t-5}^i + b - x_t^i \big\|_2^2 \tag{5}$$

gives a rotation axis and pivot if the rotation exceeds $1^\circ$ and a translation axis otherwise; with less motion the geometry channels stay empty. The part is declared blocked when

$$\beta_t = \mathbb{1}\Big[\bar{F}_t > F_0 \;\wedge\; \tfrac{1}{|M|}\textstyle\sum_{i \in M} \|x_t^i - x_{t-1}^i\| < \epsilon\Big], \tag{6}$$

with $\bar{F}_t$ the mean force over the last two steps, $F_0 = 20\,\mathrm{N}$ and $\epsilon = 0.5\,\mathrm{mm}$; a blocked part receives zero remaining travel. In the clutter scene the estimate starts from the appearance prior of each object class and is overridden by motion and by Eq. (6). In the tipping scene the estimate is $\hat{\mu} = \max(\mu_0, r/h_p)$ if the object tilted during the warm-up push, $\hat{\mu} = \min(\mu_0, r/h_p)$ if it slid, and $\hat{\mu} = \mu_0$ otherwise. The prior is $\mu_0 = 0.325$, and the center of mass keeps a fixed prior. The estimate thus adds force and segmentation to the history. The oracle separates the effect of exact physical information from estimation error.

### 3.4. World models and factor injection

**PointWorld.** PointWorld [14] lifts frozen DINOv3 [39] features to the scene points and encodes each point into a feature $e^i$. A Point Transformer V3 [53] dynamics network then predicts the displacements, taking the gripper points as additional points. We use the 51.1M-parameter configuration and train it from scratch for each setting, scene and seed. All settings of a scene use the same clips, order and number of steps (65 to 75 passes over the training clips), with batch size 128 and learning rate $10^{-4}$. The information $c_t$ enters as

$$e^i \leftarrow e^i + A_\psi\big(f_c^i \oslash \sigma\big), \tag{7}$$

where $f_c^i$ concatenates the history and factor channels of point $i$ (zero for Implicit), $\sigma$ normalizes each channel group by its root mean square on the training split, and $A_\psi$ is a two-layer perceptron. Its last layer is initialized to zero, so every setting starts from the same function and has the same number of parameters.

For the mobility factor, the predicted displacements of the part are additionally projected onto the motions that the factor allows. With $\rho^i = x_t^i - o$, its component $p^i = \rho^i - (a^\top \rho^i)a$ orthogonal to the axis and $v^i = a \times p^i$, the projection $\Pi$ replaces $\hat{\Delta}^i$ for $i \in M$ by

$$\Pi(\hat{\Delta})^i = \begin{cases} 0 & \text{if } u_t^+ = u_t^- = 0, \\ \mathrm{clip}\,\big(\bar{s}, -u_t^-, u_t^+\big)\, a & \text{prismatic geometry}, \\ (\cos\theta^\star - 1)\, p^i + \sin\theta^\star\, v^i & \text{revolute geometry}, \\ \hat{\Delta}^i & \text{no geometry}, \end{cases} \tag{8}$$

where $\bar{s} = \frac{1}{|M|}\sum_{j \in M} a^\top \hat{\Delta}^j$ is the mean predicted travel along the axis and $\theta^\star = \mathrm{clip}\,\big(\mathrm{atan2}\,\big(\sum_j w^{j\top} v^j, \sum_j w^{j\top} p^j\big), -u_t^-, u_t^+\big)$ is the rotation angle that best explains the predicted positions, with $w^j = \rho^j + \hat{\Delta}^j - (a^\top \rho^j)a$. In the clutter scene the geometry is empty and the first case applies to the points of every object marked as fixed. In the tipping scene the factor imposes no kinematic constraint and $\Pi$ is the identity. This injection was chosen among five schemes on validation clips of the storage scene before any other experiment (Sec. H). Equations (7) and (8) are the two roles of the factor that Sec. 4.2 separates.

**DINO-WM.** DINO-WM [66] encodes each frame into patch features $Z_t \in \mathbb{R}^{P\times D}$ with a frozen DINOv2 encoder [32] and predicts the next frame with a causal vision transformer,

$$\hat{Z}_{t+1} = P_\theta\big(Z_{t-K+1:t},\ \eta(\delta_{t-K+1:t}),\ \kappa([g_{t-K+1:t};\ c'])\big), \tag{9}$$

where $\eta$ and $\kappa$ embed the actions and the proprioception (the gripper position $g$) and are concatenated to every patch feature. We stack the two camera views into one image, with frames three control steps apart, and use $K = 1$ for Implicit and $K = 5$ for the history settings. The factor settings append a two-dimensional factor vector $c'$ to the proprioception. It holds the movability of the target and of its path neighbor in the clutter scene, and $\mu$ and the center-of-mass height in the tipping scene.

### 3.5. Closed-loop evaluation

**Task.** Each episode starts with five blind push steps along the inward normal of the visible face of the target, which gives the history settings their evidence. The robot must then bring the target coordinate $q$ from $q_{\text{start}}$ to $q^\star$, using at most 40 commanded increments with $\|\delta_\tau\| \le 15$ mm. Here $q$ is the joint value, or the displacement of the target along the push direction, and $q^\star - q_{\text{start}}$ is 2.5 cm for prismatic joints, 0.2 rad for revolute joints and 4 cm for the clutter and tipping scenes. A scripted push that keeps pushing into the face serves as a model-free reference.

**Planner.** With PointWorld we plan with model-predictive path integral control [52]. For a candidate sequence of increments $\delta_{1:H_p}$ over $H_p = 30$ steps, the cost is

$$\begin{aligned} J(\delta_{1:H_p}) = \frac{1}{H_p|M|}\sum_{\tau=1}^{H_p}\sum_{i\in M}\big\|\hat{x}^i_{t+\tau} - x^{i\star}\big\|_2^2 \\ + \lambda_p\sum_{\tau=1}^{H_p}\|\delta_\tau\|_2 + \lambda_r\mathcal{R}_{\text{reach}}(\delta_{1:H_p}), \end{aligned} \tag{10}$$

where $M$ is the set of target points, $x^{i\star}$ their goal positions at $q^\star$, $\lambda_p = 10^{-4}$, $\lambda_r = 0.1$, and $\mathcal{R}_{\text{reach}}$ penalizes leaving a 40 cm reach radius. Candidates are cubic splines, every iteration also scores holding still and half of the current mean, and the planner executes five steps and replans (Sec. C). With DINO-WM we use its cross-entropy-method planner [36] with an image goal $Z^\star$ rendered with the target and the gripper at their goal poses and the cost

$$J_{\text{img}} = \frac{1}{|\Omega|}\sum_{p\in\Omega}\big\|\hat{Z}_{t+H',p} - Z^\star_p\big\|_2^2 + \lambda'_p\sum_{\tau=1}^{3H'}\|\delta_\tau\|_2, \tag{11}$$

where the horizon is $H' = 5$ model steps of three control steps each. The set $\Omega = \Omega_{\text{target}}\setminus\Omega_{\text{robot}}$ contains the patches covered by the target and not by the robot in the goal image, and $\lambda'_p$ gives the path term the same share of the cost as in Eq. (10). All settings of a model share the planner and its parameters.

**Metrics.** For a free target we report the success rate $\mathbb{1}[|q_{40} - q^\star| \le \epsilon_q]$ with $\epsilon_q = 8$ mm or 0.05 rad, where overshooting counts as failure. For a locked target, and for a limited target whose stop lies short of the goal, no plan can succeed, and we report the *wasted push*

$$W = \sum_{\tau=1}^{40}\|\delta_\tau\|_2, \tag{12}$$

the total commanded gripper path; a model that recognizes the constraint stops early, and the scripted push reaches 320 mm. A model-independent probe finds the goal reachable in about one fifth of the limited articulated episodes and in about a third of the limited clutter episodes. These episodes are scored by success instead (Sec. D). In the tipping scene we report success with the object upright at the end (final tilt below 30°) and the toppling rate on tippable objects. Each scene is evaluated on 80 to 314 episodes with objects or layouts of the test split (Sec. A).

**Decision-relevant prediction error.** A planner facing a locked target mainly needs to know whether pushing will move the target. On the locked test clips we therefore measure the *phantom target motion*

$$\Phi = \frac{1}{|M|}\sum_{i\in M}\big\|\hat{\Delta}^i_{t+T}\big\|_2, \tag{13}$$

the displacement that the model predicts for a target whose true displacement is zero, which the planner can mistake for progress in Eq. (10). For tipping, the analogous quantity is $|\angle\hat{R} - \angle R|$, the error in the rotation angle of the target. Here $\hat{R}$ and $R$ solve Eq. (5) between $x_t$ and the predicted or true positions at $t + T$. To compare metrics across models we use the Pearson correlation between a single-step metric and $W$ after subtracting the mean of each scene from both.

**Controls.** Three controls act on the PointWorld models at test time only. The *test-time projection* applies $\Pi$ of Eq. (8), driven by the estimated or the oracle factor, to the predictions of an History model whose input never contained the factor. The *force stopping rule* holds the gripper still for the rest of the episode from the first replanning step at which $\beta_t = 1$ in Eq. (6), on top of the History planner or the scripted push. *Perception noise* perturbs only the inputs of the estimator. It adds Gaussian noise of 10 N to each force reading, mislabels 20% of the segmented points, or adds Gaussian noise of 1 mm to each tracked point. A fourth control retrains

History + estimated factor with the adapter of Eq. (7) alone, without the projection.

**Statistics.** In simulation, PointWorld results average three training seeds and three planner seeds, and DINO-WM results two training seeds. The controls use the first planner seed and are compared with the same runs of the main models. Per episode we average over seeds and test the mean paired difference to History with a bootstrap over episodes. We mark differences whose 95% interval excludes zero, and averages over scenes are tested by resampling episodes within each scene.

# 4. Results

## 4.1. Explicit mobility changes decisions where the target cannot move

Table 1 reports single-step prediction and closed-loop control for the five mobility scenes. Averaged over the four scenes other than the sliding window, the estimated factor lowers the moving-point error on free clips from 7.44 to 7.16 mm relative to History, a change of 4%. In closed loop the History + estimated factor setting pushes 245 mm on average before stopping, against 391 mm for History and 417 mm for Implicit, and the difference to History is significant in all four scenes. On limited targets whose goal lies beyond the stop the wasted push falls from 359 to 277 mm, significantly in the articulated scenes but not in the clutter scene (414 to 396 mm). Success on free targets rises from 62% to 67% on average, mainly in storage furniture (48% to 66%) and appliance doors (59% to 66%), while the table drawer loses 4 points. The oracle factor removes the wasted push on locked targets almost entirely, because the output projection holds a part with zero remaining travel still (Sec. 4.2).

The sliding window shows what happens when history already identifies the factor. A sash moves along a single rail, so five frames of a push reveal its state. The History model predicts a phantom motion of only 0.27 mm, and the estimated factor does not shorten the push significantly.

## 4.2. Two roles of the factor

The factor reaches PointWorld as an input to the point encoder (Eq. (7)) and as a projection of the predicted motion (Eq. (8)). Table 2 separates the two on the five mobility scenes. The History + estimated factor models shorten the push against locked targets by 122 mm relative to History. Applied at test time to History models that never received the factor, the estimated projection alone shortens it by 107 mm, and the oracle projection removes it entirely. Retraining History + estimated factor with the input alone shortens it by 54 mm, and on unreachable limited targets by 19 of 76 mm. A network that is free to use the factor thus learns only part of the constraint that the projection imposes.

The two mechanisms differ on free targets. Factor conditioning raises success over History by 4.9 points with the projection and by 3.2 points without it. The test-time projection alone adds 1.5 points, which is not significant. The benefit of conditioning beyond the projection alone is 3.5 points (interval $[0.7, 6.4]$), mostly in the storage scene.

## 4.3. Stopping does not need a world model

The stopping behavior relies on the same evidence as the estimator, a high contact force on a target that does not move. We therefore compare it with a rule that acts on this evidence directly. Added to the History planner, the force stopping rule wastes 42 mm on locked targets and 112 mm on unreachable limited targets. This is less than every learned setting without oracle information (Tab. 2), and success on free targets stays at the level of History. The decision to stop thus follows from contact sensing alone. The rule does not improve the pushing of movable targets, which is where factor conditioning helps.

The estimator tolerates 10 N of force noise and 20% of mislabeled part points. Noise of 1 mm on the tracked points exceeds the stillness threshold of Eq. (6), and the models then push 37 mm farther than History (Tab. 12).

## 4.4. Stopping gains persist with scaling

In the clutter scene, a 411M-parameter dynamics network trained on five times as many clips lowers the moving-point error of History from 6.86 to 5.13 mm (Tab. 13). The scaled History model still pushes 470 mm against fixed targets, against 450 mm for the small one, and the scaled History + estimated factor model 218 mm less (95% interval $[197, 236]$).

## 4.5. What predicts closed-loop behavior

Figure 3 compares three single-step errors as predictors of the wasted push on locked targets across the 60 mobility models, with scene means removed. The standard moving-point error, computed over all test clips, correlates with the wasted push at $r = 0.56$, the static-point error on locked clips at $r = 0.64$, and the phantom motion of the target at $r = 0.77$. The advantage of the phantom motion over the standard error is 0.21 (95% interval $[0.10, 0.33]$ under a bootstrap over models). Without the 15 oracle models the ordering is the same (0.58, 0.60, 0.66) but the advantage of 0.08 has an interval that includes zero. On the four ambiguous scenes the estimated factor removes 40% of the phantom motion that history leaves (3.5 to 2.1 mm). This matches its 37% reduction of the wasted push, while the standard error moves by 4%. In Fig. 4 the model without history lets a locked part follow the gripper, and the factor settings hold it still.

Table 1. Mobility scenes, PointWorld, test split. Single-step errors are in millimeters: moving points of free clips, static points of locked clips, and the phantom motion of the target on locked clips. Closed-loop columns give the success rate on free targets and the wasted push on locked targets and on limited targets whose goal lies beyond the stop. Values average three training seeds and three planner seeds; episode counts are listed in Table 6. †: the paired 95% interval of the difference to History excludes zero.

| | | Single-step error (mm) ↓ | | | Success (%) ↑ | Wasted push (mm) ↓ | |
|---|---|---|---|---|---|---|---|
| Scene | Setting | Moving, free | Static, locked | Phantom, locked | Free | Locked | Limited |
| Storage | Implicit | 10.32 | 0.331 | 10.18 | 60† | 393† | 371† |
| | History | 6.53 | 0.175 | 4.82 | 48 | 322 | 285 |
| | + estimated factor | 6.17 | 0.099 | 2.82 | 66† | 227† | 216† |
| | + oracle factor | 5.62 | 0.075 | 0.01 | 66† | 1† | 80† |
| | *scripted push* | | | | 13 | 320 | 320 |
| Table drawer | Implicit | 13.59 | 0.208 | 8.57 | 79† | 442 | 443 |
| | History | 8.94 | 0.119 | 3.29 | 86 | 456 | 430 |
| | + estimated factor | 8.85 | 0.078 | 1.95 | 82† | 269† | 284† |
| | + oracle factor | 9.67 | 0.049 | 0.02 | 84 | 0† | 92† |
| | *scripted push* | | | | 7 | 320 | 320 |
| Appliance door | Implicit | 10.52 | 0.221 | 6.80 | 63 | 346 | 339† |
| | History | 7.42 | 0.106 | 2.28 | 59 | 336 | 308 |
| | + estimated factor | 7.20 | 0.090 | 1.51 | 66 | 220† | 212† |
| | + oracle factor | 6.50 | 0.061 | 0.00 | 68 | 0† | 76† |
| | *scripted push* | | | | 8 | 320 | 320 |
| Sliding window | Implicit | 13.71 | 0.118 | 2.69 | 27 | 324† | 454† |
| | History | 11.72 | 0.052 | 0.27 | 23 | 174 | 246 |
| | + estimated factor | 11.96 | 0.057 | 0.22 | 29† | 150 | 192 |
| | + oracle factor | 11.81 | 0.040 | 0.05 | 34† | 0† | 108† |
| | *scripted push* | | | | 14 | 320 | 320 |
| Clutter | Implicit | 8.35 | 0.191 | 7.87 | 61† | 486† | 432† |
| | History | 6.86 | 0.149 | 3.64 | 54 | 450 | 414 |
| | + estimated factor | 6.42 | 0.104 | 2.15 | 54 | 265† | 396 |
| | + oracle factor | 5.77 | 0.098 | 0.00 | 57† | 0† | 422 |
| | *scripted push* | | | | 3 | 321 | 320 |

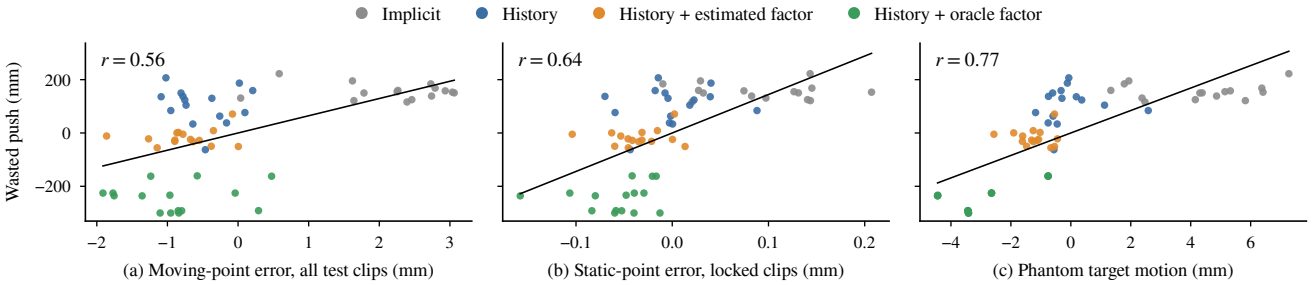



Figure 3. Which single-step error tracks closed-loop behavior on locked targets. Each point is one trained PointWorld model (five mobility scenes, four settings, three seeds). The horizontal axis is (a) the standard moving-point error over all test clips, (b) the static-point error on locked clips, and (c) the phantom motion of the target on locked clips. The vertical axis is the wasted push on locked episodes. Both axes are centered within each scene.

### 4.6. Sliding versus toppling

Table 3 reports the tipping scene. History lowers the prediction error on topple clips from 28.6 to 21.6 mm, but the toppling rate on tippable objects stays at 36%. Both factors lower it, to 31.7% with the estimated and to 29.9% with the oracle factor, while success with the object upright is maintained (56.8% and 55.7% against 55.5%). The estimate separates sliding from tilting episodes rather than recovering

Table 2. Two roles of the factor and a model-free alternative, PointWorld, mean over the five mobility scenes (first planner seed). Success on free targets (%), wasted push on locked and on unreachable limited targets (mm). †: the 95% interval of the difference to History, bootstrapped within scenes, excludes zero. Per-scene values in Table 12.

| Setting | Free ↑ | Locked ↓ | Limited ↓ |
|---|---|---|---|
| History | 53.8 | 346 | 332 |
| History + estimated factor | 58.7† | 223† | 255† |
| without projection | 57.0† | 292† | 313† |
| History + estimated projection at test time | 55.3 | 239† | 262† |
| History + oracle projection at test time | 57.3† | 1† | 138† |
| History + force stopping rule | 54.0 | 42† | 112† |

Table 3. Tipping scene, PointWorld, test split. Single-step errors on slide and topple clips (moving points, mm) and the error of the predicted rotation of the target on topple clips (degrees). Closed loop: success with the object upright and the toppling rate on objects able to tip (%). † as in Table 1.

| | Single step ↓ | | | Closed loop | |
|---|---|---|---|---|---|
| Setting | Slide | Topple | Rot. | Upright ↑ | Topple ↓ |
| Implicit | 6.25 | 28.63 | 21.7 | 52.7 | 35.2 |
| History | 5.55 | 21.55 | 18.1 | 55.5 | 35.9 |
| + est. factor | 5.96 | 20.09 | 16.4 | 56.8† | 31.7† |
| + oracle factor | 5.58 | 18.82 | 15.8 | 55.7 | 29.9† |

Table 4. DINO-WM in closed loop, test split, two training seeds. Columns as in Tables 2 and 3. Of 200 episodes per scene, those completed by every run are scored (196 in the clutter scene). † as in Table 1.

| | Clutter | | | Tipping | |
|---|---|---|---|---|---|
| Setting | Free ↑ | Locked ↓ | Limited ↓ | Upright ↑ | Topple ↓ |
| Implicit | 22.8 | 345† | 322† | 19.0† | 24.2† |
| History | 27.2 | 392 | 354 | 32.8 | 32.1 |
| + est. factor | 32.6 | 408 | 372 | 33.0 | 37.3 |
| + oracle factor | 25.5 | 404 | 349 | 32.8 | 32.9 |

the friction (Sec. B). With no projection to act through, its effect comes entirely from factor conditioning, the role that also improves pushing of movable targets (Sec. 4.2).

### 4.7. An image-feature world model

Table 4 repeats the comparison with DINO-WM, where the factor is only an input. No factor setting differs significantly from History in either scene. In the clutter scene the oracle and estimated settings push 404 and 408 mm against a fixed target, as far as History (392 mm). In the tipping scene all three history settings keep the object upright in 33% of the episodes. The factor changes predictions without improving closed-loop stopping. Flipping the movability of the target changes the predictions on locked clips by more than the predicted change of the scene, and the true factor predicts the next frame better (Sec. E). Without a projection, the factor enters the decision only through the learned dynamics, which in PointWorld also recover only part of the constraint (Sec. 4.2).

Table 5. Real-robot closed loop, 10 locked and 10 free layouts. $W$: wasted push on locked targets (mm, mean), counted from the first warm-up step. Lower: layouts with less push than History. A false stop ends more than 8 mm from the goal with the last ten commands zero.

| | Locked | | Free | |
|---|---|---|---|---|
| Setting | $W$ ↓ | Lower | Success ↑ | False stop ↓ |
| History | 195 | – | 5/10 | 0/10 |
| + estimated projection at test time | 152 | 8/10 | 5/10 | 0/10 |
| + estimated factor | 130 | 8/10 | 7/10 | 0/10 |
| + force stopping rule | 40 | 10/10 | 5/10 | 0/10 |

### 4.8. Real-robot evaluation

We repeat the stopping comparison on a real arm (Tab. 5; setup in Sec. J). An SO-101 arm pushes a coffee capsule among other capsules, a real counterpart of the clutter scene. Double-sided tape under its base locks the target, and a free target stands on the same tape with the liner on. The models receive a 4,096-point cloud from monocular depth in two RGB views and motion history from tags on the capsule lids. The elbow servo load replaces the contact force of Eq. (6). History and History + estimated factor are fine-tuned from their clutter checkpoints on 36 scripted pushes recorded on the arm.

We compare four settings of Tab. 2: History, the estimated projection applied at test time, History + estimated factor, and the force stopping rule. All use the MPPI planner of Sec. 3.5 on 10 locked and 10 free layouts. After a warm-up push of five 8 mm steps, the arm must move the target 4 cm within 40 steps in total. On locked targets we report the wasted push $W$ of Eq. (12), counted from the first warm-up step.

On locked targets History pushes 195 mm on average. Applied to the same model at test time, the projection shortens this by 43 mm to 152 mm, and History + estimated factor shortens it by 65 mm to 130 mm. Both push less than History in 8 of the 10 layouts. The force stopping rule pushes only the 40 mm of the warm-up in every layout. On free targets History + estimated factor reaches the goal in 7 of 10 layouts and each other setting in 5 of 10, and no setting stops falsely. As in simulation, the output constraint shortens the push against locked targets, the contact rule stops earliest, and History + estimated factor reaches the most free goals.

## 5. Conclusion

Explicit physical information helps a manipulation world model beyond observation history in ways that depend on how the information is used. Constraining the predicted motion improves decisions about blocked objects, often without retraining the model. Conditioning the learned dynamics gives smaller, scene-dependent gains on movable objects and reduces toppling of tall objects. A simple contact rule resolves the stopping decision better than the learned predictions. Making physics explicit is thus most useful where it changes a decision the model would otherwise get wrong. The same ordering of stopping behavior appears on a real arm.

## References


[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[2] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 1, 2

[3] Chengyu Bai, Peidong Jia, Tiecheng Guo, Yukai Wang, Rui Ma, Fangyuan Zhao, Chunkai Fan, Xiaobao Wei, Jintao Chen, Hao Wang, Ying Li, Xiaozhu Ju, Jian Tang, and Shanghang Zhang. Ioi: Decoupling kinematics and physics for interactive world models. *arXiv preprint arXiv:2606.23296*, 2026. 2

[4] Peterson Co, Sicheng Hu, Chunxuan Jiao, Hongyang Cheng, Yulin Luo, Yijie Xu, Sixiang Chen, Zhongxia Zhao, Zihao Wang, DaFeng Chi, Peidong Liu, YuTong Chen, Henghua Liu, Zhihao Yuan, Huizhu Jia, Yuzheng Zhuang, Tianle Zhang, Liang Lin, Huajie Tan, and Shanghang Zhang. Worldsimprobe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. *arXiv preprint arXiv:2608.09298*, 2026. 2

[5] Mohamad H. Danesh, Chenhao Li, Amin Abyaneh, Anas Houssaini, Kirsty Ellis, Glen Berseth, Marco Hutter, and Hsiu-Chin Lin. Morphology-conditioned world model for cross-embodiment quadrupedal locomotion. *arXiv preprint arXiv:2604.08780*, 2026. 2

[6] Bardienus P. Duisterhof, Kaifeng Zhang, Adam Hung, Bowen Wen, Stan Birchfield, Yunzhu Li, Deva Ramanan, and Jeffrey Ichnowski. Pointzero: 3d point track completion for learning transferable 3d dynamics. *arXiv preprint arXiv:2609.19142*, 2026. 2

[7] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. In *Robotics: Science and Systems (RSS)*, 2022. 2

[8] Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: Object-centric world models for robotics manipulation. *arXiv preprint arXiv:2307.02427*, 2023. 2

[9] Koyo Fujii, Luis Figueredo, Praminda Caleb-Solly, Ivan Boschi, Edoardo Ida', Marco Carricato, and Aly Magassouba. Phypush: One push is all you need for sensorless physical property estimation with physics-guided transformers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. 2

[10] Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[11] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019. 1, 2

[12] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 1, 2

[13] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[14] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 1, 2, 4

[15] Steven M. Hyland, Jing Xiao, and Cagdas D. Onal. Multimodal non-prehensile estimation of physical parameters via press-and-pull tipping. *arXiv preprint arXiv:2609.33138*, 2026. 2

[16] Steven M. Hyland, Jing Xiao, and Cagdas D. Onal. Before the tipping point: Force-guided active perception for shape-agnostic estimation of 3d centers of mass. *arXiv preprint arXiv:2609.12894*, 2026. 2

[17] Taoran Jiang, Yixuan Guan, Liqian Ma, Jing Xu, Jiaojiao Meng, Weihang Chen, Zecui Zeng, Lusong Li, Dan Wu, and Rui Chen. Dexsim2real$^2$: Building explicit world model for precise articulated object dexterous manipulation. *arXiv preprint arXiv:2409.08750*, 2024. 2

[18] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[19] Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Richards, and Mike Rabbat. Interpreting physics in video world models. *arXiv preprint arXiv:2602.07050*, 2026. 2

[20] Keon Woo Kim. Latent state design for world models under sufficiency constraints. *arXiv preprint arXiv:2605.01694*, 2026. 2

[21] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021. 1, 2

[22] Abhinav Kumar, Soshi Iba, Rana Soltani Zarrin, and Dmitry Berenson. Plume: Probabilistic latent unified world modeling and parameter estimation for multi-finger manipulation. *arXiv preprint arXiv:2606.11396*, 2026. 2

[23] Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Learning for Dynamics and Control Conference (L4DC)*, 2020. 1, 2

[24] Jiajian Li, Jingyuan Huang, Junru Gong, Qi Wang, Xiaokang Yang, and Yunbo Wang. Orbisim: World models as differentiable physics engines for embodied intelligence. *arXiv preprint arXiv:2605.16395*, 2026. 1, 2

[25] Wenxuan Li, Hang Zhao, Zhiyuan Yu, Yu Du, Qin Zou, Ruizhen Hu, and Kai Xu. Pin-wm: Learning physics-informed world models for non-prehensile manipulation. In *Robotics: Science and Systems (RSS)*, 2025. 2

[26] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[27] Ruiyao Liu, Boshu Lei, Zhuoyang Pan, and Kostas Daniilidis. Learning to explore hidden kinematics for articulated object manipulation. *arXiv preprint arXiv:2609.36553*, 2026. 2

[28] Youngjae Min and Navid Azizan. Hardnet: Hard-constrained neural networks with universal approximation guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 14

[29] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[30] Thijmen Nijdam. Local world models for articulated object manipulation. Master's thesis, University of Amsterdam, 2025. https://thijmennijdam.github.io/local-world-models/. 2

[31] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 15

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 5

[33] Kei Ota, Hsiao-Yu Tung, Kevin A. Smith, Anoop Cherian, Tim K. Marks, Alan Sullivan, Asako Kanezaki, and Joshua B. Tenenbaum. H-saur: Hypothesize, simulate, act, update, and repeat for understanding object articulations from interactions. *arXiv preprint arXiv:2210.12521*, 2022. 2

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 14

[35] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018. 14

[36] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999. 5

[37] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, 2020. 2

[38] Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. In *Robotics: Science and Systems (RSS)*, 2022. 2

[39] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 4

[40] Kaizhen Tan, Sizhe Xu, Xin Xu, Siru Tao, Yixiao Li, Hanzhe Hong, Yang Feng, Heqing Du, and Zhaonan Wang. What can latent world models know? physical information in multimodal predictive representations. *arXiv preprint arXiv:2607.27017*, 2026. 2

[41] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. In *Robotics: Science and Systems (RSS)*, 2025. 1, 3

[42] Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2025. 2

[43] Adam J. Thorpe, Stepan Tretiakov, Cheng-Hsi Hsiao, Su Ann Low, Xingjian Li, Hassan Iqbal, Neel P. Bhatt, Ufuk Topcu, and Krishna Kumar. Physically viable world models: A case for query-conditioned embodied ai. *arXiv preprint arXiv:2605.30542*, 2026. 2

[44] Claas Voelcker, Victor Liao, Animesh Garg, and Amir massoud Farahmand. Value gradient weighted model-based reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022. 2

[45] Linhao Wang, Yiyan Fan, and Dongjin Huang. Not all errors matter: Decision-relevant prediction error predicts planning quality. *arXiv preprint arXiv:2609.32322*, 2026. 1, 2

[46] Meizhong Wang, Wanxin Jin, Kun Cao, Lihua Xie, and Yiguang Hong. Contactgaussian-wm: Learning physics-grounded world model from videos. *arXiv preprint arXiv:2602.11021*, 2026. 2

[47] Maggie Wang, Stephen Tian, Aiden Swann, Ola Shorinwa, Jiajun Wu, and Mac Schwager. Phys2real: Fusing vlm priors

with interactive online adaptation for uncertainty-aware sim-to-real manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. 2

[48] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 15

[49] Xi Wang, Tianxing Chen, Qiaojun Yu, Tianling Xu, Zanxin Chen, Yiting Fu, Ziqi He, Cewu Lu, Yao Mu, and Ping Luo. Articulated object manipulation using online axis estimation with sam2-based tracking. *arXiv preprint arXiv:2409.16287*, 2024. 2

[50] Yufeng Wang, Parivesh Priye, Lu Wei, and Haibin Ling. Conservation buys stability and factoring buys counterfactuals in physical world models. *arXiv preprint arXiv:2609.19674*, 2026. 2, 14

[51] Ran Wei, Nathan Lambert, Anthony McDonald, Alfredo Garcia, and Roberto Calandra. A unified view on solving objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2310.06253*, 2023. 2

[52] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016. 2, 5

[53] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[54] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[55] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B. Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In *Robotics: Science and Systems (RSS)*, 2019. 2

[56] Xu Yang, Yiqin Yang, and Qianchuan Zhao. Graph-operator world models for morphology-parameter generalization in continuous control. *arXiv preprint arXiv:2608.20936*, 2026. 2

[57] Hantao Ye, Ross Worobel, Zhuoli Xie, Mingen Li, Houjian Yu, Youngjin Hong, and Changhyun Choi. Pointcast: One world model for rigid, articulated, and deformable object manipulation. *arXiv preprint arXiv:2609.28393*, 2026. 2

[58] Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems (RSS)*, 2017. 2

[59] Yang Yu, Shiyuan Zhang, Yifei Sheng, Haoxiang Ren, and Haoxin Lin. How should world models be evaluated for embodied decision-making? a decision-making-centric position. *arXiv preprint arXiv:2606.15032*, 2026. 2

[60] Guanqi Zhan, Xianzheng Ma, Weidi Xie, and Andrew Zisserman. Inferring dynamic physical properties from video foundation models. *arXiv preprint arXiv:2510.02311*, 2025. 2

[61] Jiaming Zhang and Homanga Bharadhwaj. Track, articulate, act: Generating articulation from casual human videos. *arXiv preprint arXiv:2609.19119*, 2026. 2

[62] Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations (ICLR)*, 2026. 1, 2

[63] Jiaqi Zhang, Feng Ye, Mingjia Yang, Zhihong Chen, Mingkang Xiang, Xinglin Yao, Yanbin Li, Siwei Ma, and Chuanmin Jia. Robotwin-phys: Do wams and vlas understand the physical world? *arXiv preprint arXiv:2609.26292*, 2026. 2

[64] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024. 1, 2

[65] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 14

[66] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, 2025. 1, 2, 5

## A. Data

Table 6 lists the size of each split. Objects (PartNet-Mobility models for the articulated scenes, generated layouts for the clutter and tipping scenes) never appear in more than one split. Each clip contains the current frame, five history frames and ten future frames recorded by two RGB-D cameras at $320 \times 180$ pixels, with 4,096 scene points tracked through the clip by the simulator and 400 points sampled on the gripper. In each articulated episode the pushed part is free with probability one half and locked or limited with probability one quarter each. All main evaluation results are reported on the test split, whose objects or layouts never appear in training or validation, and each test episode is run once per model and planner seed. A validation split of further held-out objects or layouts was used to choose the injection scheme (Sec. H) and the DINO-WM planning cost (Sec. E); these validation results and the estimator diagnostics are reported separately in this appendix.

Table 6. Data sizes. Objects or layouts and clips for training and testing, and the number of closed-loop test episodes per hidden state that finished in every run of the scene.

| | Objects or layouts | | Clips | | Episodes | | |
|---|---|---|---|---|---|---|---|
| Scene | Train | Test | Train | Test | Free | Locked | Limited |
| Storage | 205 | 46 | 18,327 | 3,685 | 157 | 79 | 78 |
| Table drawer | 50 | 10 | 17,177 | 5,839 | 119 | 61 | 59 |
| Appliance door | 37 | 8 | 19,354 | 3,340 | 40 | 20 | 20 |
| Sliding window | 31 | 6 | 13,717 | 2,932 | 50 | 27 | 24 |
| Clutter | 4739 | 1020 | 18,622 | 4,018 | 90 | 20 | 86 |
| Tall object | 4900 | 1050 | 19,599 | 4,199 | | 200 (all) | |

**Scripted pushes.** The data-collection script plans a collision-free approach to a face of the target that is visible from the cameras, closes the gripper, and pushes along the inward normal of that face with small directional noise. It never reads the joint state or the hidden factor. After the first push segment it chooses each further segment among pushing at 4 to 12 mm per step (45%), slow pushing (15%), pausing (15%), retreating (15%) and side steps (10%). The action given to the model is the commanded gripper trajectory, which keeps advancing when the target is blocked.

## B. Factor estimation

**Mobility.** The estimator receives the five history frames of the scene points, the segmentation of the part, and the contact force between the gripper and the part over the history. If at least 16 points of the part moved by more than 2 mm, it fits a rigid transform between the first and last history frames and classifies the motion as a rotation when the rotation angle exceeds $1^\circ$, recovering the axis and pivot, and as a translation otherwise. It marks the part as blocked when the mean contact force over the last two history steps exceeds 20 N while the part moved less than 0.5 mm in the last step, and then sets the remaining travel to zero in the direction of the observed motion, or in both directions if the part never moved. Otherwise the remaining travel is set to a cap of 0.3 m. When the part has not moved, the geometry channels stay empty and only the part mask and the travel are filled.

**Clutter.** The estimate starts from the class prior (items movable, fixtures fixed). An object that moved by more than 3 mm over the history is set to movable, and the target is set to fixed when the contact force exceeds 20 N while it stays still. On validation episodes the estimate of the target agrees with the truth in 96% of episodes at the first planning step and in 81% at the last, where blocked limited targets are sometimes read as fixed.

**Tipping.** If the target tilted by more than $3^\circ$ during the history, the friction is set to the larger of the prior (0.325) and the ratio of the half width to the push height, which is the friction at which a push at that height starts a tilt. If it slid without tilting, the friction is set to the smaller of the two. The height of the center of mass is always the prior value of 40% of the object height. The resulting friction estimate separates sliding from tilting episodes and correlates with the true friction at $r = 0.31$ over the 4,199 test clips.

## C. Training and planning details

Table 7 lists the parameters of each model and Tab. 8 the settings used for training, planning and estimation. All settings of a model share these values; the four information settings differ only in the inputs and the projection described in Secs. 3.3 and 3.4.

Table 7. Model components and parameter counts. The factor adapter is present in every PointWorld setting, so all settings of a model have the same number of parameters. The DINO-WM predictor has 20.0M parameters with $K = 1$ and 20.3M with $K = 5$; its image decoder is trained but not used for planning.

| Model | Component | Parameters | Trained |
|---|---|---|---|
| PointWorld (S) | DINOv3 ViT-L encoder | 303.2M | no |
| | Dynamics network | 51.1M | yes |
| | Factor adapter $A_\psi$ | 22.7k | yes |
| PointWorld (B) | DINOv3 ViT-L encoder | 303.2M | no |
| | Dynamics network | 411.0M | yes |
| | Factor adapter $A_\psi$ | 39.2k | yes |
| DINO-WM | DINOv2 ViT-S/14 | 22.1M | no |
| | Predictor | 20.0–20.3M | yes |
| | Action, proprio. embeddings | <0.2k | yes |
| | Image decoder | 10.1M | yes |

Table 8. Training, planning and estimation settings.

| | |
|---|---|
| *PointWorld training* | |
| Optimizer | AdamW, learning rate $10^{-4}$ (constant), weight decay 0.01, gradient clipping at norm 5 |
| Batch | 16 clips on each of 8 GPUs (128) |
| Steps | 10,000 (8,000 for the sliding window); 65 to 75 passes over the training clips |
| Seeds | 3 training seeds per setting and scene |
| Clip | 1 current, 5 history and 10 future frames; 4,096 scene points, 400 gripper points; two cameras at $320 \times 180$ |
| *MPPI planner (PointWorld)* | |
| Horizon | 30 steps, predicted in three model calls of 10 steps |
| Sampling | 32 samples, 3 iterations, cubic splines through knots at steps 0, 10, 20, 30; knot noise 3 cm, decay 0.6 per iteration |
| Weights | temperature $\beta = 0.02$ times the cost range; $\lambda_p = 10^{-4}$, $\lambda_r = 0.1$, reach radius 40 cm |
| Execution | 5 steps per replan, 8 replans (40 steps); planner seeds 0, 1, 2 |
| *DINO-WM training and CEM planner* | |
| Training | batch 32; predictor and embeddings at $5 \times 10^{-4}$, decoder at $3 \times 10^{-4}$; 8 epochs (Implicit) or 15 to 16 epochs (history settings), about 30k steps |
| Input | two views stacked into one $224 \times 224$ image; frames 3 control steps apart; $K = 1$ (Implicit) or $K = 5$ |
| CEM | 300 samples, top 30, 30 iterations, horizon 5 model steps, 2 model steps executed per replan |
| Path cost | $\lambda_p'$ set so that a steady 8 mm push over the horizon costs 2% of the initial goal cost |
| *Task and estimator* | |
| Task | 5 warm-up push steps, at most 40 commanded steps of at most 15 mm; success within 8 mm or 0.05 rad |
| Mobility estimator | rigid fit on at least 16 points moved by more than 2 mm; rotation above $1^\circ$; blocked if $\bar{F}_t > 20$ N and part motion below 0.5 mm |

**PointWorld.** We use the released code and objective without its learned-uncertainty term. The step count is the number of steps for 60 passes over the training clips, rounded up to a multiple of 2,000. Normalization statistics are computed once per scene on the training split and shared by all settings, and the final checkpoint is evaluated without selection.

**MPPI.** The planner optimizes gripper translation with the orientation fixed. Before each of the three model calls the factor is recomputed at the predicted state: the oracle travel is reduced by the predicted motion along the axis, and the estimate is recomputed from the predicted history. Every iteration also scores holding still and half of the current mean, and the executed plan is the cheaper of the final weighted mean and the best sample. The plan is shifted for warm starting.

**DINO-WM.** Both camera views are encoded as JPEG at quality 90, stacked vertically and letterboxed. Action and proprioception are embedded to 10 dimensions and concatenated to every patch feature, and the factor is appended to the three-dimensional gripper position before embedding. The action is the commanded gripper increment of each of the three control steps between frames. A training window holds $K$ context frames and the next frame; at test time, history frames that precede the recorded episode repeat its first frame.

## D. Limited episodes

In the articulated scenes the stop of a limited part is drawn from a fixed random stream when the episode is created, so it can be recomputed exactly for every episode; no recorded joint position of any run leaves the recomputed stop. The goal of a limited episode is reachable when it lies inside the stop up to the success tolerance. In the clutter scene the fixed neighbor can deflect the target sideways and no closed form exists, so reachability is decided by a probe that does not involve any evaluated model: from the same initial state, scripted pushes along the push direction rotated by $-20^\circ$, $-10^\circ$, $0^\circ$, $10^\circ$ and $20^\circ$ in the table plane advance 8 mm per step for 40 steps, and the episode counts as reachable when any of them brings the target within tolerance of its goal. Table 9 gives, on the test split, the share of reachable limited episodes, the success rate on them, and the commanded gripper path over all limited episodes.

Table 9. Limited episodes. $n$: limited episodes; Reach.: share with a reachable goal (%); success on the reachable ones (%) and commanded gripper path over all limited episodes (mm) for Implicit, History, History + estimated factor and History + oracle factor.

| | | | Success, reachable | | | | Path, all limited | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Scene | $n$ | Reach. | I | H | E | O | I | H | E | O |
| Storage | 78 | 22 | 49 | 33 | 61 | 71 | 343 | 272 | 215 | 106 |
| Table drawer | 59 | 10 | 56 | 67 | 67 | 67 | 424 | 407 | 281 | 101 |
| Appliance door | 20 | 15 | 93 | 96 | 100 | 96 | 307 | 282 | 201 | 91 |
| Sliding window | 24 | 29 | 13 | 11 | 14 | 14 | 398 | 210 | 191 | 124 |
| Clutter | 86 | 31 | 66 | 53 | 55 | 54 | 382 | 368 | 350 | 372 |

## E. Factor use and planning cost in DINO-WM

**Does the model use the factor?** For 160 held-out slices of the clutter scene, we predicted the next model step once with the true factor and once with the movability of the target flipped, keeping images, history and actions fixed. Table 10 reports the change between the two predictions relative to the change the model predicts for the scene, and the prediction

errors against the observed next frame. On locked slices the flip changes the prediction by more than the predicted scene change itself, and in every state the true factor predicts the next frame better than the flipped one.

Table 10. Counterfactual factor check, DINO-WM, clutter, seed 0. Effect: mean squared feature difference between predictions with the true and the flipped factor, relative to the predicted scene change. Errors: mean squared feature error against the observed next frame ($\times 10^3$) with the true factor, the flipped factor, and a prediction that nothing changes.

| Model | State | $n$ | Effect | True | Flipped | Static |
|---|---|---|---|---|---|---|
| + oracle factor | free | 70 | 0.32 | 104.7 | 120.2 | 135.0 |
| | limited | 56 | 0.29 | 87.8 | 92.7 | 102.1 |
| | locked | 34 | 1.54 | 70.9 | 84.8 | 84.0 |
| + estimated factor | free | 70 | 0.29 | 103.7 | 118.4 | 135.0 |
| | limited | 56 | 0.27 | 86.8 | 92.3 | 102.1 |
| | locked | 34 | 1.01 | 73.3 | 79.9 | 84.0 |

**Planning cost.** The planner of Sec. 3.5 differs from the original DINO-WM planner in three respects, fixed on validation episodes before any test run. The cost excludes the patches covered by the robot in the goal image, it penalizes the commanded path, and it scores holding still and half of the mean at every iteration. Without these changes, moving the gripper towards its goal pose lowers the image cost even when the target does not move, and all settings push against locked clutter targets for 460 to 500 mm. Table 11 reports all settings under the original cost. Under either cost, the explicit factors do not shorten the push against fixed targets.

Table 11. DINO-WM with the original planning cost (image goal including the robot, no path penalty, no hold candidate), validation episodes. Columns as in Table 4, with all limited episodes in the Limited column.

| | Clutter | | | Tipping | |
|---|---|---|---|---|---|
| Setting | Free ↑ | Locked ↓ | Limited ↓ | Upright ↑ | Topple ↓ |
| Implicit | 29.5† | 461 | 435† | 37.0 | 24.6† |
| History | 52.3 | 474 | 452 | 40.2 | 32.0 |
| + est. factor | 53.8 | 486 | 455 | 40.8 | 34.4 |
| + oracle factor | 43.2 | 499† | 462 | 41.2 | 41.4† |

## F. Controls per scene

Table 12 lists the controls of Secs. 4.2 and 4.3 for each mobility scene, with the first planner seed and three training seeds. The stopping rule and the test-time estimated projection shorten the push against locked targets significantly in every scene. The retrained input-only setting does so in three scenes, by less than History + estimated factor with the projection, is indistinguishable from History in the storage scene, and pushes farther than History in the sliding window, where history already resolves the factor. The estimator tolerates 10 N of force noise and 20% of mislabeled part points. Noise of 1 mm on the tracked points exceeds the stillness threshold of Eq. (6), and the models then push 37 mm farther than History (Tab. 12).

## G. Scaling

The scaled model uses the PTv3-B configuration of the released code (predictor width 256, 411M dynamics parameters) and the clutter training set enlarged from 18,622 to 93,350 clips with new layouts generated by the same procedure, with the normalization statistics of the original data and the same procedure for the step count (44k steps, about 60 passes). One training seed is run per setting. Table 13 lists the results next to the 51M-parameter models.

## H. Injection schemes

The way a factor is injected changes its effect [34, 50], so we compared five schemes for the oracle mobility factor in the storage scene with training seed 0 on validation clips and episodes (Tab. 14): additive adapters with a normalized or a zero-initialized output [65], conditional normalization [34, 35], joint tokens, and the zero-initialized adapter with the output projection [28]. A rule fixed before training selected the scheme on single-step errors alone; the projection gave the lowest errors. The closed-loop columns, computed afterwards, anticipate Sec. 4.2: with the four schemes that use the factor only as an input, the model given the true factor still pushes between 115 and 351 mm against a locked door.

## I. Selection of the qualitative examples

The examples of Fig. 4 use the seed-0 models of the tables and were chosen by rules fixed before any prediction was rendered. Each prediction panel shows the first locked held-out clip, in shard order, in which the gripper advances by more than 2 cm and the contact force at the current frame exceeds 20 N.

## J. Real-robot implementation and evaluation

### J.1. Hardware and perception

Figure 5 shows the setup and the input of the model. This section follows the input in the order in which it is computed: camera images, depth and point cloud, tag motion history, servo load and factor estimate.

**Arm and objects.** The arm is an SO-101 follower arm with STS3215 servos in position control. Its position gain

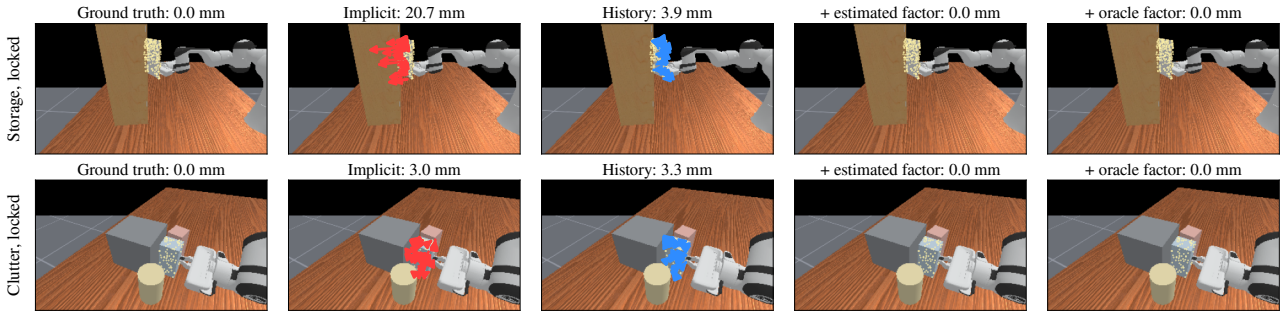



Figure 4. Predicted motion of the target over ten steps on locked held-out clips (arrows magnified six times; titles give $\Phi$ of Eq. (13)). The true motion is zero, so every arrow is phantom motion. Seed-0 models; clips chosen by the rules of this section.

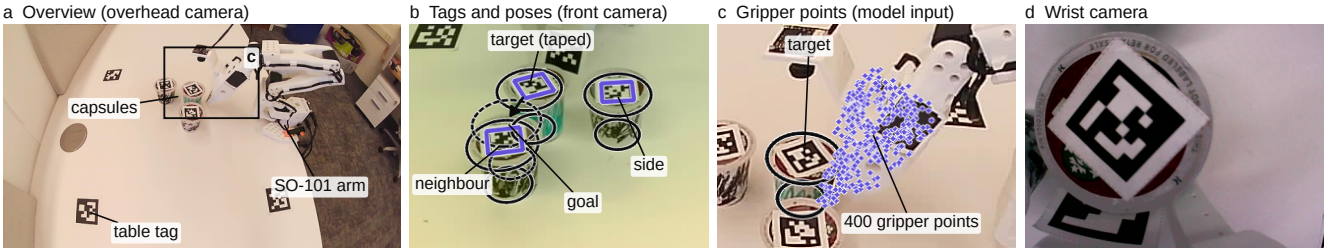



Figure 5. Real-robot setup and model input. (a) Overhead camera: the SO-101 arm, the tagged capsules and the table tags; the box marks the region of (c). (b) Front camera: tag detections (violet) and tracked capsule poses (black rings); the arrow points from the target to its goal 4 cm ahead (dashed). (c) The 400 gripper points given to the model at the end of the warm-up. (d) Wrist camera, used only for recording.

leaves small commands up to $2.7^\circ$ short, so each step is re-commanded until the measured joints are within $0.4^\circ$ of their targets. The arm pushes with the sides of its closed jaws, leaning $25^\circ$ forward. The lowest gripper point starts 12 mm above the table and meets a capsule about 22 mm up its wall. The objects are six coffee capsules, 50 mm wide at the top, 35 mm at the base and 42 mm tall, each with a 23.7 mm AprilTag [31] on its lid. A locked capsule stands on a pad of double-sided tape, and a free capsule stands on the same pad with its liner on.

**Cameras.** A front camera with a pinhole model and an overhead camera with a fisheye model give the input. A wrist camera only records. The camera poses and the joint zero offsets of the arm are fitted jointly to tag observations. The table plane is taught with the gripper.

**Point cloud.** MoGe-2 [48] predicts metric depth in each input view. Each depth map is rescaled to the calibrated table plane, and the two views are fused in the robot base frame. The fused cloud is cropped to the workspace, cleared of robot points by forward kinematics, voxelized at 5 mm and sampled to 4,096 points. The 400 gripper points are placed by forward kinematics of the measured joint angles.

**Motion history.** The tag poses give the motion of each capsule, from which the history $H_t$ of Eq. (3) is formed; depth builds only the current point cloud. For the stillness test of Eq. (6), each camera intersects its ray through the tag center with the lid plane. A still capsule then moves by 0.05 to 0.09 mm per step in the median, well below $\epsilon = 0.5$ mm.

**Contact signal and factor estimate.** The elbow servo load, averaged over two steps, takes the place of $\bar{F}_t$ in Eq. (6). Its threshold $F_0 = 200$ is in units of 0.1% of full scale. On 21 free and 21 locked straight pushes outside the fine-tuning data, the load rose after contact by at most 7 and by at least 144. With $F_0 = 200$, Eq. (6) flagged every locked push within two steps of contact and no free push. The factor is then estimated as in the clutter scene (Sec. B): every capsule starts as movable, and at each replan the target is set to fixed when Eq. (6) holds.

## J.2. Model adaptation

History and History + estimated factor start from their seed-0 checkpoints of the clutter scene, with the 51.1M-parameter dynamics network. For fine-tuning we recorded 36 scripted push episodes on the arm in three blocks of 12, each with 7 free and 5 locked episodes. An episode has the five warm-up steps and 30 further steps whose segments follow the proportions of the simulation data (Sec. A). One episode per

Table 12. Controls per scene, PointWorld, test split, first planner seed. Success on free targets (%), wasted push on locked and on unreachable limited targets (mm). † as in Table 1.

| Setting | Free ↑ | Locked ↓ | Limited ↓ |
|---|---|---|---|
| *Storage* | | | |
| History | 46 | 318 | 284 |
| History + estimated factor | 66† | 227† | 217† |
| without projection | 60† | 312 | 298 |
| History + estimated projection at test time | 52† | 223† | 228† |
| History + oracle projection at test time | 54† | 0† | 64† |
| History + force stopping rule | 47 | 31† | 80† |
| Est. factor, force noise 10 N | 68† | 240† | 238† |
| Est. factor, 20% segmentation errors | 58† | 262† | 256† |
| Est. factor, tracking noise 1 mm | 60† | 368† | 333† |
| Scripted push + stopping rule | 13 | 27 | 45 |
| *Table drawer* | | | |
| History | 88 | 452 | 430 |
| History + estimated factor | 83† | 267† | 282† |
| without projection | 86 | 289† | 301† |
| History + estimated projection at test time | 84† | 312† | 308† |
| History + oracle projection at test time | 86 | 2† | 98† |
| History + force stopping rule | 86 | 68† | 119† |
| Est. factor, force noise 10 N | 86 | 273† | 295† |
| Est. factor, 20% segmentation errors | 77† | 281† | 313† |
| Est. factor, tracking noise 1 mm | 82† | 431† | 451 |
| Scripted push + stopping rule | 5 | 60 | 91 |
| *Appliance door* | | | |
| History | 58 | 334 | 302 |
| History + estimated factor | 66 | 218† | 206† |
| without projection | 64 | 252† | 216† |
| History + estimated projection at test time | 62 | 224† | 200† |
| History + oracle projection at test time | 68 | 0† | 40† |
| History + force stopping rule | 56 | 25† | 51† |
| Est. factor, force noise 10 N | 60 | 231† | 232† |
| Est. factor, 20% segmentation errors | 55 | 219† | 229† |
| Est. factor, tracking noise 1 mm | 47 | 287† | 271 |
| Scripted push + stopping rule | 10 | 38 | 28 |
| *Sliding window* | | | |
| History | 25 | 173 | 226 |
| History + estimated factor | 26 | 156 | 179 |
| without projection | 24 | 222† | 338† |
| History + estimated projection at test time | 25 | 133† | 200 |
| History + oracle projection at test time | 26 | 1† | 60† |
| History + force stopping rule | 29 | 82† | 70† |
| Est. factor, force noise 10 N | 27 | 158 | 200 |
| Est. factor, 20% segmentation errors | 25 | 139† | 162† |
| Est. factor, tracking noise 1 mm | 35 | 297† | 390† |
| Scripted push + stopping rule | 14 | 160 | 165 |
| *Clutter* | | | |
| History | 51 | 451 | 415 |
| History + estimated factor | 53 | 248† | 391† |
| without projection | 51 | 384† | 412 |
| History + estimated projection at test time | 54 | 301† | 375† |
| History + oracle projection at test time | 53 | 0† | 426† |
| History + force stopping rule | 52 | 4† | 238† |
| Est. factor, force noise 10 N | 54 | 246† | 383† |
| Est. factor, 20% segmentation errors | 54 | 268† | 388† |
| Est. factor, tracking noise 1 mm | 56† | 529† | 455† |
| Scripted push + stopping rule | 6 | 2 | 203 |

block is held out for validation, which leaves 33 episodes for training. Table 15 lists the fine-tuning settings. Both settings use the same data, schedule and learning rate, chosen on the validation moving-point error of History. The test-time projection and the force stopping rule act on the fine-tuned History model without further training.

Table 13. Scaling PointWorld in the clutter scene, test split. Moving-point error on free clips (mm), success on free targets (%), and wasted push on locked and unreachable limited targets (mm). PTv3-S rows average three training seeds, PTv3-B rows use one; all rows average three planner seeds.

| Model | Setting | Error ↓ | Free ↑ | Locked ↓ | Limited ↓ |
|---|---|---|---|---|---|
| PTv3-S, 1× data | History | 6.86 | 53.5 | 450 | 414 |
| | + est. factor | 6.42 | 54.3 | 265 | 396 |
| PTv3-B, 5× data | History | 5.13 | 58.5 | 470 | 373 |
| | + est. factor | 5.91 | 59.6 | 252 | 386 |

Table 14. Injection schemes for the oracle mobility factor, storage scene, seed 0, validation split. Moving-point error on free clips and static-point error on locked clips (mm); closed-loop success (%) and wasted push on all locked and limited episodes (mm).

| Injection | Moving | Static | Success | Locked | Limited |
|---|---|---|---|---|---|
| Additive, normalized | 8.66 | 0.080 | 64 | 115 | 241 |
| Additive, zero init. | 6.51 | 0.026 | 69 | 169 | 262 |
| Conditional norm. | 7.02 | 0.076 | 62 | 130 | 231 |
| Joint tokens | 7.36 | 0.132 | 58 | 351 | 289 |
| Zero init. + projection | 6.05 | 0.013 | 67 | 0 | 70 |

Table 15. Fine-tuning on the real arm. History and History + estimated factor use the same values.

| | |
|---|---|
| Initialization | seed-0 clutter checkpoint of the same setting |
| Data | 36 scripted episodes (21 free, 15 locked); 33 for training and 3 for validation; windows every 2 steps |
| Clip | as in simulation: 1 current, 5 history and 10 future frames; 4,096 scene points and 400 gripper points |
| Normalization | statistics recomputed on the real training episodes |
| Optimizer | AdamW as in simulation (Tab. 8): weight decay 0.01, gradient clipping at norm 5; constant learning rate $10^{-5}$ |
| Batch and steps | 16 clips on one GPU, the per-GPU batch of simulation; 1,000 steps |
| Selection | learning rate in $\{10^{-5}, 3\cdot10^{-5}, 10^{-4}\}$ and 1k, 2k or 4k steps, by the validation moving-point error of History |

## J.3. Evaluation protocol

**Layouts and reset.** A program draws each layout. The target lies 22 to 29 cm from the robot base within $\pm30^\circ$ of the forward direction, and its goal lies 4 cm ahead in the vertical plane of the arm. A path neighbor stands ahead of the target, next to one or two side capsules and up to two distractors. The operator places the capsules, guided by their tags, until each lies within 3 mm and $2^\circ$ of its drawn pose. The evaluation uses 10 layouts with a locked target and 10 with a free one. Each layout is run once with every setting

and reset to its drawn poses before each run.

**Execution.** Each episode starts with five warm-up steps of 8 mm along the push direction. MPPI then plans with the cost and parameters of Sec. 3.5 and Tab. 8 and executes five steps of at most 15 mm between replans. The gripper moves at most 2 cm backwards or sideways, and its commanded lowest point stays at least 5 mm above the table. The episode ends after 40 steps, warm-up included, or once the target has moved 8 cm. The force stopping rule holds the arm still for the rest of the episode from the first planning step at which $\beta_t = 1$.

**Metrics.** A free episode succeeds when the target ends within 8 mm of its goal in the table plane after the gripper retreats, so a target pushed past its goal fails. A false stop is a free episode that ends more than 8 mm from its goal with its last ten commands zero. On locked targets $W$ sums the length of every commanded step from the first warm-up step, so the warm-up contributes 40 mm. A setting pushes less than History on a layout when its $W$ is lower than that of History on the same layout.