I FEEL HOW YOU SEE: A DYNAMICS WORLD MODEL FOR PREDICTING FORCES AND KINEMATICS FOR HIGH-TOUCH ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In contact-rich manipulation, forces and torques are the fundamental drivers of contact dynamics, dictating how objects move and react to manipulation. Yet, most world models today forgo the grounding that comes from predicting plausible force/torque (F/T) traces alongside their predicted robot trajectories. Addressing this gap, we present a Bin Dynamics World Model, *BDWM*, for robotic bin stowing via space-creating sweep motions. Conditioned solely on pre-contact information (an RGB-D image, item segmentation, and the planned blade sweep), a single flow-matching velocity field jointly generates two latent streams: a physics latent that decodes to the full F/T and end-effector trajectory, and a vision latent that predicts the post-contact scene. Evaluated on real-robot cycles, *BDWM* reduces trajectory CRPS by $\sim$75% (0.009 vs. 0.036) over the strongest generative action-model baseline and trajectory RMSE by $\sim$26% (0.025 vs. 0.034 m) over the strongest baseline overall. Lightweight heads on the predicted latents further drive downstream tasks such as the prediction of space created after a sweep, reaching sub-centimeter accuracy compared to a task-specific regression baseline.

1 INTRODUCTION

World models have become central to learning-based control as predictors of how an environment evolves under an agent's action. All of them model the visual future: they forecast pixels or visual latents and couple this to a policy (Lyu et al., 2025; Sun et al., 2026; Su et al., 2026a; Zheng et al.; Hafner et al., 2020). However, in contact-rich manipulation, the decisive variable is not just what the scene will look like but what the robot will feel: the force/torque (F/T) signature of an interaction is the clearest evidence of whether contact proceeded as intended and how the interaction unfolded. Force is increasingly used as an input to contact-rich policies (Yu et al., 2025), and a few recent models predict future wrenches as an execution-time reference (He et al., 2026). However, a high-fidelity world model that predicts the physical consequences of an action (forces, motion, and resulting scene) before execution from pre-contact information alone remains open. We study this in the context of large-scale ($\sim$250K) confined space multi-item manipulation using data from the robotic system of (Hudson et al., 2025), where a blade is inserted into a cluttered bin (Figure 1) and swept to displace items and create space for a new item to be stowed (we use pre-stow and post-stow for the pre- and post-contact phases throughout). The outcome hinges on the partially observable configuration of items the blade will contact; thus, the same visible scene and planned sweep can yield very different force traces and outcomes: a clean sweep, a jam, or a dropped item. This mapping from (scene, plan) to realized dynamics is, thus, an inherently multimodal use of a generative world model: we must represent a distribution over what could happen, not a single expected trace. We present *BDWM*, a factored latent dynamics world model that jointly generates the contact dynamics and post-stow scene of a sweep using only pre-stow inputs. Our contributions are:

(1) **Bin Dynamics World Model (BDWM).** We introduce a generative world model for contact-rich stowing that jointly generates the contact dynamics trace and post-stow scene latent from pre-stow inputs via conditional flow matching trained and evaluated on various manipulation data. We introduce a force-weighted spatio-temporal grounding loss that aligns the physics tokens' attention with the pre-stow image patches the blade touches, and demonstrate through ablations that co-generating

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

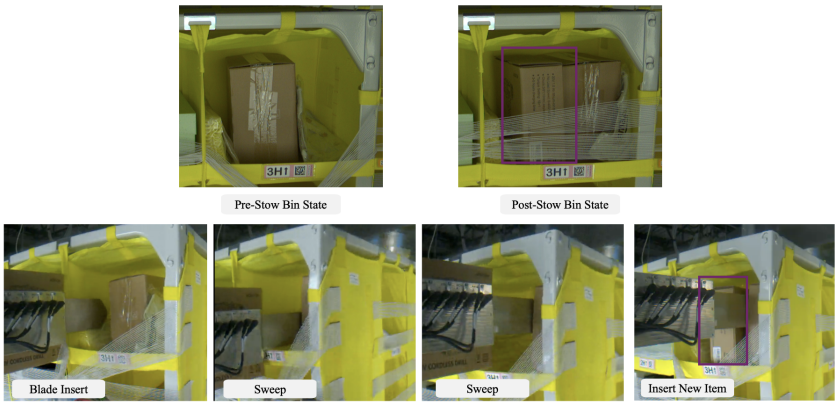

Figure 1: **Blade-sweep stowing task.** Top: pre- and post-contact RGB views of the bin, before and after the robot blade sweep. Bottom: the blade is inserted into a packed bin and swept to displace items and open space for a new one. As the items the blade will contact might be occluded from the pre-stow view, similar-looking bins can yield very different force traces and outcomes, motivating a generative model of contact dynamics.

the post-stow scene and grounding the image to the physics each yield measurable gains in the predicted traces.

(2) **We show that co-generating F/T and trajectory improves motion-prediction fidelity.** *BDWM* leverages a single velocity field that jointly transports the force/torque and trajectory latent, such that the predicted wrench conditions the motion rather than being emitted as an isolated channel. Ablating the F/T head and generating motion alone degrades the trajectory RMSE from 0.025 to 0.094 m, a larger regression than removing any single conditioning input (Table 2), providing direct evidence that the coupling carries the trajectory gain.

(3) **BDWM produces informative latents for downstream tasks, such as predicting the space created by a sweep.** Lightweight heads on the frozen *BDWM* latents predict the created space, providing pre-execution features for stow-success and sweep-outcome prediction. More broadly, this suggests a path for world models in which the sampled traces are used not only to roll out futures but also as feature extractors for downstream control.

2 Related Work

Force in manipulation learning. Force/torque sensing is a critical modality for identifying when and where contact occurs in contact-rich tasks; however, much prior work has underexploited it. Several methods consume F/T as an observation to improve contact-rich policies (Yu et al., 2025; Li et al., 2026; Zhao et al., 2026), where the measured wrenches condition the policy to modulate contact. However, fewer methods predict F/T jointly with the robot's action or trajectory, despite the wrench being a causal driver of the resulting motion; we show that co-generating the two yields more accurate action and trajectory predictions. The closest work to ours, (He et al., 2026), jointly predicts future actions and end-effector wrenches to model contact evolution, using the predicted wrenches as an execution-time reference for closed-loop residual correction. We differ in timing and in what the prediction is for: their wrench conditions control after contact begins, whereas we generate the full distribution of contact dynamics before contact from pre-contact inputs alone when the occluded configuration still admits many outcomes and reads out its spread rather than a single reference trace. Our predicted distribution acts as a pre-execution uncertainty signal whose spread tracks stow outcome, a use of predicted forces not addressed by prior force-aware policies, motivated by outcome prediction at the decision-making step to help select high-level manipulation strategies, not behavior shaping at the execution.

World models and world-action models. A large body of work models how visual observations evolve under action and couples this prediction with control, either as explicit world models or as world-action models and vision-language-action (VLA) policies (Lyu et al., 2025; Su et al., 2026b; Sun et al., 2026; Black et al., 2024; Physical Intelligence et al., 2025). In model-based RL, a learned

latent dynamics model is rolled out to plan or to train a policy in imagination, as in the Dreamer line of work (Hafner et al., 2020), establishing latent world models as a method for control. Recent analysis argues that much of the benefit of video prediction in such models comes from training-time representation learning rather than test-time future generation (Zheng et al.). Our model predicts a physical signal (force and motion) along with visual latents rather than pixels, hypothesizing that jointly predicting physics and vision latents captures dynamics information useful for downstream tasks. We draw on this literature for the encoder and conditioning design but target a different question: whether a pre-execution generative model of contact dynamics yields an input-dependent, decision-relevant uncertainty signal.

Generative action policies and dynamics. Robotic bin manipulation involves item sorting, bin rearrangement, and sweeping/pushing to create space, typically studied through grasp or motion planning over observed geometry (Ali et al., 2018; Cheong et al., 2020; Ding et al., 2023; Schwarz et al., 2018). Diffusion and flow-matching-based action heads are now standard for capturing such multimodal data distributions in manipulation policy learning (Black et al., 2024; Physical Intelligence et al., 2025), building on rectified-flow and score-based generative modeling (Liu et al., 2022; Song et al., 2020). We adopt a flow-matching head to represent the multimodal dynamics of contact, where the same visible scene and plan admit many outcomes, and to obtain a sample-based uncertainty estimate for our stowing task. Unlike these action generators, our velocity field transports coupled physics and vision latents jointly, and we read out its sample spread as a predictive-uncertainty signal rather than sampling a single action to execute.

3 Problem Statement

We use data from the system in (Hudson et al., 2025), which stows heterogeneous items into flexible fabric pods and bins at e-commerce fulfillment centers. The manipulator is a six-DOF arm on a two-axis gantry (2+6-DOF), with a thin, extendable aluminum blade at its end-effector, and is instrumented with a six-axis force/torque (F/T) sensor. To make room for a new item, the blade is inserted into a cluttered bin and swept along a space-creation axis (Figure 1) under an admittance controller that pushes with a target force. The sweep thus probes available space and item compressibility through kinematic feedback. The trajectory is selected from 3D cost-map affordances computed on the pre-stow scene. Control runs at 500 Hz; the F/T signal and end-effector trajectory are logged at 50 Hz over each $\sim$4 s sweep, synchronized with pre- and post-stow RGB-D and item segmentation imagery. The conditioning image comes from a fixed pedestal camera that is occluded during manipulation, so live images are unavailable at deployment. We repurpose this data for *BDWM*, predicting a planned sweep's F/T-plus-trajectory trace and resulting visual configuration from pre-contact inputs alone.

4 Method

We model the sweep as a conditional generative model, instantiated as factored latent flow matching: a single velocity field v_θ jointly transports two latent streams, a physics latent z_{phys} that decodes to the contact-dynamics trace, and a vision latent z_{vis} that encodes the post-stow image. Its flow transports Gaussian noise $z_0 \sim \mathcal{N}(0, I)$ to samples from $p(z_{\text{phys}}, z_{\text{vis}} \mid o, a)$, conditioned on the pre-stow observation o and planned sweep a; every conditioning signal is strictly pre-contact, and the post-stow image and ground-truth motion enter only as training targets. Decoding $[z_{\text{phys}};\ z_{\text{vis}}]$ with $\texttt{decoder}_{\texttt{ft}}$ recovers the F/T-plus-trajectory trace τ, while z_{vis} is supervised toward $z_{\text{vis}}^{\text{tgt}}$, the DINOv3 (Siméoni et al., 2025) embedding of the real post-stow image. We supervise the vision stream in this frozen embedding space rather than in pixels as reconstruction would spend capacity on appearance (item textures, labels, lighting) that does not bear on the contact dynamics whereas DINOv3 patches retain the spatial layout of the items the sweep displaces and share the feature space of the pre-stow conditioning (Zheng et al.). We define the conditioning architecture, the two latent spaces, and the training objective in (Section 4.1), (Section 4.2) and (Section 4.3), respectively.

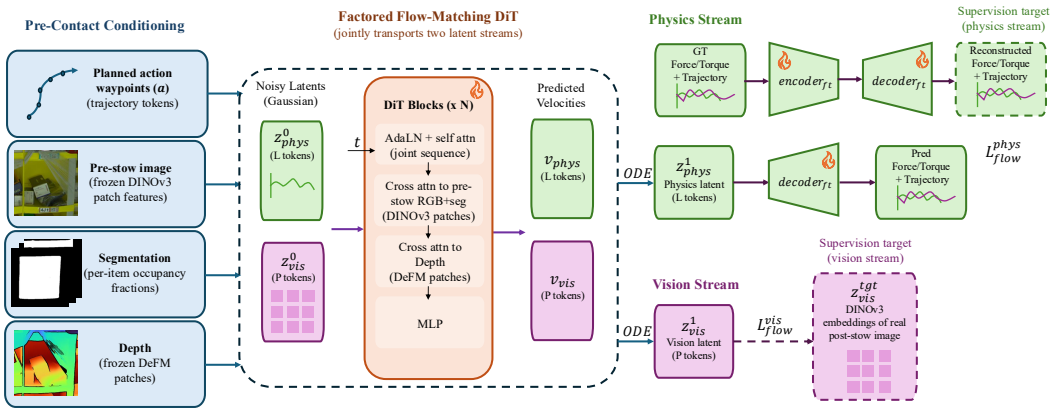

Figure 2: **BDWM overview.** A factored flow-matching DiT, conditioned only on pre-contact signals (frozen DINOv3 pre-stow image features, segmentation, depth, and the planned sweep), jointly generates a physics latent and a post-stow vision latent; the physics latent decodes to the force/torque and trajectory trace, while the vision latent is supervised toward the frozen DINOv3 embedding of the real post-stow image.

4.1 Conditioning: A factored cross-attention DiT

The velocity field v_θ is a diffusion transformer (DiT) (Peebles & Xie, 2023) that operates on a single token sequence: the two latent streams it transports ($z_{\text{phys}}, z_{\text{vis}}$) become the *query* tokens, and every conditioning signal enters as either extra sequence tokens or cross-attention key/value banks. The noisy z_{phys} (L tokens) and z_{vis} (P tokens) are embedded and concatenated with the planned trajectory tokens into the joint sequence $[z_{\text{phys}};\ z_{\text{vis}};\ \text{waypoints}]$ using three input streams:

(1) **Pre-stow RGB (with segmentation).** Frozen DINOv3 ViT-S (Siméoni et al., 2025) patches of the pre-stow image, linearly projected to model width, form an RGB key/value bank. Per-patch, per-item segmentation occupancy fractions are projected and added into this same bank, giving the RGB cross-attention per-patch item-awareness at no extra token cost. This segmentation projection is zero-initialized, so item-awareness is introduced gradually over training.

(2) **Depth.** DeFM (Patel et al., 2026) depth patches ($d_{\text{depth}} = 1024$) form a second key/value bank, sharing the RGB spatial encoding (below), thus grounding both to identical bin-frame locations. The depth pathway begins as a no-op with zero initialization and is added gradually over training.

(3) **Planned sweep waypoints.** The pre-contact planned action waypoints a are embedded as trajectory tokens, one per timestep, and placed directly in the query sequence alongside z_{phys} and z_{vis}.

DiT block. The model is a stack of N identical blocks; the joint sequence (physics, vision, and waypoint rows) passes through all of them, and the RGB and depth banks are cross-attended in every block, so conditioning is injected at every layer rather than once. Each block applies, in order, (i) AdaLN-modulated self-attention over the joint query sequence; (ii) a cross-attention sublayer into the RGB bank; (iii) a second, separate cross-attention sublayer into the depth bank with its own AdaLN modulation; and (iv) an MLP. Readout happens only after the final block: zero-initialized linear heads on the last-layer hidden states of the physics and vision rows produce the per-token velocities v_{phys} and v_{vis}, while the waypoint rows are carried through the stack as pure context and then discarded (there is no separate decoder). The network thus realizes the joint velocity field $v_\theta(z_{\text{phys}}, z_{\text{vis}}, t \mid o, a) = (v_{\text{phys}}, v_{\text{vis}})$; the latents are never emitted directly but recovered by integrating this field from Gaussian noise at $t{=}0$ to $t{=}1$, while at training the heads regress the target velocities of eqs. (1) and (2) below.

Positional encodings. To let cross-attention align *where* the blade contacts the scene with *when* it does so, we apply RoPE (Su et al., 2024) along two axes in a shared bin-frame coordinate system defined at the bin face as the Y-axis right-to-left and the Z-axis upward. Spatially, each RGB and depth key/value patch is rotated by the physical $(Y_{\text{bin}}, Z_{\text{bin}})$ coordinate of the surface it images, giving the image tokens an explicit metric identity in the same frame as the blade-tip motion. Temporally, the physics and waypoint tokens are rotated by their timestep index, $\theta_k = \pi\,\text{base}^{-2k/d_h}$ with per-head dimension $d_h = 64$ and base $= T/2 = 100$; the π factor pins the fastest rotary pair to a 2-token (40 ms) period. Ablations in Table 8.

4.2 LATENT REPRESENTATIONS

Physics latent and motion autoencoder. The physics target z^1_{phys} lives in a latent that preserves the trace's native temporal resolution. The realized trace concatenates the F/T signal ($C_{\text{FT}} = 13$) and observed end-effector trajectory ($C_{\text{traj}} = 17$) channel-wise, giving $C = 30$ channels at 50 Hz over $T = 200$ steps (20 ms/step, a 4s sweep (Appendix Table 5 and Table 6 for details of the channels used). A 1-D convolutional motion autoencoder maps this $30 \times T$ trace through three stride-1 padded convolutions, emitting one token per raw timestep, so $z_{\text{phys}} \in \mathbb{R}^{B\times L\times 256}$ with $L = T = 200$; encoding the ground-truth trace yields the physics flow target z^1_{phys}. Keeping the latent at the native rate rather than compressing to a global bottleneck is deliberate: a force spike stays localized in one token instead of being pooled across the sequence. A transposed stride-1 convolutional decoder, $\texttt{decoder}_{\texttt{ft}}$, maps the physics latent, concatenated with the pooled vision latent at decode time (Section 4.3), back to the $30 \times T$ trace. The autoencoder is trained jointly with the flow network, so the latent stays decodable while its geometry is shaped by the flow-matching loss into a space the velocity field can model well.

Post-stow vision latent (future-state target). The vision target z^1_{vis} is a frozen embedding of the post-stow image, so transporting toward it forces the shared latent to explain how the bin will look after the sweep. A frozen DINOv3 ViT-S encoder (embedding dim $d_v = 384$) applied to the post-stow RGB image yields L_2-normalized patches, giving the vision flow target $z^1_{\text{vis}} = z^{\text{tgt}}_{\text{vis}} \in \mathbb{R}^{P\times d_v}$ with $P = 256$.

4.3 TRAINING OBJECTIVE

Both latent streams, z_{phys} and z_{vis}, are trained with a rectified-flow (linear-interpolant) velocity objective (Liu et al., 2022). For a stream with target z_1 and noise $z_0 \sim \mathcal{N}(0, I)$, the interpolant is $z_t = (1-t)z_0 + tz_1$ and the network regresses the constant target velocity $z_1 - z_0$ (detached targets), giving one loss per stream: one on the physics latent, one on the vision latent:

$$\mathscr{L}^{\text{phys}}_{\text{flow}} = \left\| v^{\text{phys}}_\theta - (z^{\text{phys}}_1 - z^{\text{phys}}_0) \right\|^2, \tag{1}$$

$$\mathscr{L}^{\text{vis}}_{\text{flow}} = \left\| v^{\text{vis}}_\theta - (z^{\text{vis}}_1 - z^{\text{vis}}_0) \right\|^2, \qquad z^{\text{vis}}_1 = z^{\text{tgt}}_{\text{vis}} \tag{2}$$

The flow time follows a logit-normal schedule, $t = \sigma(\varepsilon)$ with $\varepsilon \sim \mathcal{N}(0,1)$, concentrating training on intermediate noise levels. Two decoder terms operate on the $30 \times T$ trace. Both feed the motion decoder the concatenation $[z^{\text{phys}};\ \text{pooled}\, z^{\text{vis}}]$, where z^{vis} is mean-pooled over its P patches and broadcast onto every physics token (decoder input dim $256 + d_v$). A reconstruction term decodes the ground-truth-encoded latents $[z^{\text{phys}}_1;\ \text{pooled}\, z^{\text{vis}}_1]$ with decoder gradients on:

$$\mathscr{L}_{\text{recon}} = \frac{1}{\sum_i m_i} \sum_i m_i \left\| \text{decoder}_{\text{ft}}\left([z^{\text{phys}}_1;\ \text{pooled}\, z^{\text{vis}}_1]\right)_i - \tau^{\text{GT}}_i \right\|^2, \tag{3}$$

with $m_i \in \{0, 1\}$ masking invalid timesteps.

Peak-Weighted Loss. A peak-weighted term instead decodes the flow's one-step endpoint estimate $\hat{z} = z_t + (1-t)\, v_\theta$ through the gradient-detached decoder, so its gradient flows into the flow network. It is the same masked MSE against τ^{GT} as $\mathscr{L}_{\text{recon}}$ but with each step weighted by $w_i = 1 + \alpha\, |\tau^{\text{GT}}_i| / \text{scale}$ ($\alpha = 4$, scale the per-channel std of τ^{GT}), so large-magnitude (force-spike) steps dominate (full equation in Section A.2).

Spatio-temporal grounding. This term couples *what* the model attends to in the pre-stow scene to *when* and *how hard* the blade makes contact. A grounding loss $\mathscr{L}_{\text{ground}}$ supervises the physics→RGB cross-attention of the last block, $A \in \Delta^N$ per physics token (each row an attention distribution over the N image patches). For each physics token i (a timestep), we form a soft spatial target $t_{i,n}$ concentrated on the image patches nearest the commanded blade position $p_i = (Y, Z)$ (a Gaussian over bin-frame patch coordinates, $\sigma = 0.05$ m, masking patches with no valid depth), and push the attention toward it with a per-token cross-entropy $\text{CE}_i = -\sum_n t_{i,n} \log A_{i,n}$. Each token is weighted by its normalized blade contact-force magnitude w_i (clipped to $[f_{\min}, 1]$, $f_{\min} = 0.1$), so high-force contact timesteps dominate and free-space timesteps contribute little: $\mathscr{L}_{\text{ground}} = \left(\sum_i w_i\, \text{CE}_i\right) / \sum_i w_i$. This ties the spatial pre-contact representation to the temporal dynamics the model must generate (soft-target and weight definitions in Section A.3; ablations in Table 9).

Putting the four terms together, the training objective is

$$\mathscr{L} = \lambda_{\text{phys}}\mathscr{L}_{\text{flow}}^{\text{phys}} + \lambda_{\text{vis}}\mathscr{L}_{\text{flow}}^{\text{vis}} + \lambda_{\text{ft}}\mathscr{L}_{\text{recon}} + \lambda_{\text{peak}}\mathscr{L}_{\text{peak}} + \lambda_{\text{ground}}\mathscr{L}_{\text{ground}}, \tag{4}$$

with $\lambda_{\text{phys}} = \lambda_{\text{vis}} = 1$, $\lambda_{\text{ft}} = 3$, $\lambda_{\text{peak}} = 0.005$, and λ_{ground} linearly warmed up from 0 over the first 20 epochs to a target of 0.02.

5 Experiments and Results

5.1 Experimental Setup

Inference Protocol. We integrate the joint flow from $t = 0$ to 1 with forward Euler in 150 steps. Our main sampler is a Stochastic Differential Equation (SDE) (Song et al., 2020) with churn 0.5 (Ablation on SDE vs. ODE (Chen et al., 2018) in Table 10). Classifier-free guidance with weight g_{cfg} interpolates the conditional and null velocities, $v = v_{\text{null}} + g_{\text{cfg}}(v_{\text{cond}} - v_{\text{null}})$. After integration, z_{vis} is mean-pooled over its P patches and concatenated with z_{phys}, and $\texttt{decoder}_{\texttt{ft}}$ decodes $[z_{\text{phys}};\ \text{pooled } z_{\text{vis}}]$ into the predicted trace, split into F/T (13 channels) and trajectory (17 channels). Since sampling is stochastic (the initial latent z_0 and any injected SDE noise vary across draws) for each pre-contact input (o, a) we draw K samples $(z_{\text{phys}}^{(k)}, z_{\text{vis}}^{(k)})_{k=1}^{K}$ (default $K = 50$).

Metrics. We evaluate two complementary properties: the calibration and sharpness of the predictive distribution, and the fidelity of individual samples to the ground truth. For each channel, we pool all valid (timestep, cycle) pairs; every point carries K i.i.d. model samples with an ensemble mean μ_n and empirical std σ_n, scored against the ground truth y_n and averaged over channels. We report four scores (full definitions in Section A.5). **CRPS** (ensemble/energy estimator (Lang et al., 2026)) is a score that rewards accurate, appropriately-dispersed forecasts, penalizing both over- and under-dispersion. **GP-NLL** scores the predictive under a Gaussian process over the full trace, capturing joint temporal calibration (autocorrelation across the sweep) rather than treating timesteps independently; a per-element Gaussian variant, which agrees on all rankings, is in Section A.5. Both are Gaussian summaries, and CRPS remains our non-parametric score sensitive to multimodality. **RMSE** of the ensemble mean measures point accuracy. **best-of-K** (min-RMSE over the K per-cycle rollouts) is an oracle-selection score of whether any sampled rollout matches the realized dynamics, rewarding a distribution that covers the true outcome even when its mean does not.

Baselines. Our baseline methods allow us to ask two questions: whether a policy or action-model architecture repurposed for dynamics matches a purpose-built contact-dynamics world model and whether generative modeling is needed at all versus deterministic regression.
Regression: We compare against a deterministic regression variant that takes the same inputs as *BDWM* and outputs either the F/T or trajectory using fully connected layers.

$\boldsymbol{\pi_0}$ (Black et al., 2024) and $\boldsymbol{\pi_{0.5}}$ (Physical Intelligence et al., 2025): Flow-matching vision-language-action policies

ForceVLA (Yu et al., 2025): A force-aware VLA with a force-conditioned mixture-of-experts.

All four were designed to generate actions rather than to model contact dynamics, so we adapt each to our setting: we train it on the same split to regress the sweep trace from identical pre-contact conditioning (pre-stow RGB-D, segmentation, and the planned sweep). Crucially, π_0, and $\pi_{0.5}$ do not model force, so they are evaluated only on the trajectory channels; ForceVLA takes force as input but does not predict force, and the regression baseline models force and is scored on both modalities. Results in Table 1 and Section 5.2.

Ablations. Table 2 ablates the two architectural axes (co-generated post-stow vision stream $\mathscr{L}_{\text{flow}}^{\text{vis}}$, F/T head) and two conditioning modalities (pre-stow image, sweep waypoints); every variant shares the split, isolating one design choice.

5.2 Main Results

Result 1 - Joint F/T-trajectory generation attains the highest accuracy on both modalities. *BDWM* dominates every metric on both modalities (Table 1). Critically, no baseline predicts both

Table 1: Probabilistic trace prediction on 24,783 held-out cycles (K=50, SDE sampler). Deterministic models produce a single trace. Their distributional scores are marked "dnmu" - does not model uncertainty. π_0, $\pi_{0.5}$ and ForceVLA do not model force (F/T entries marked "dnm" - does not model force).

	Force/Torque				Trajectory			
Method	CRPS ↓	GP-NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	GP-NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
Regression baseline	dnmu	dnmu	10.213	dnmu	dnmu	dnmu	0.0561	dnmu
π_0 (Black et al., 2024)	dnm	dnm	dnm	dnm	0.037	13.52	0.068	0.027
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	dnm	dnm	dnm	dnm	0.036	14.75	0.067	0.026
ForceVLA (Yu et al., 2025)	dnm	dnm	dnm	dnm	dnmu	dnmu	0.034	dnmu
Ours	**3.19**	**3.54**	**5.75**	**4.12**	**0.009**	**−4.78**	**0.025**	**0.005**

Table 2: Architectural and conditioning ablations that evaluate each design choice.

	Force/Torque				Trajectory			
Method	CRPS ↓	GP-NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	GP-NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
No post-stow ($\mathscr{L}_{\text{flow}}^{\text{vis}}$)	4.78	3.58	9.29	6.13	0.024	4.32	0.056	0.029
No image conditioning	4.56	3.82	8.59	6.08	0.043	3.58	0.093	0.030
No action conditioning	4.25	7.16	8.08	5.45	0.037	−2.63	0.084	0.012
No F/T prediction	dnm	dnm	dnm	dnm	0.024	−1.8	0.094	0.064
Ours	**3.19**	**3.54**	**5.75**	**4.12**	**0.009**	**−4.78**	**0.025**	**0.005**

from a shared generative process: π_0 and $\pi_{0.5}$ emit trajectory only, whereas ForceVLA consumes force as an observation but does not predict it. *BDWM*'s negative trajectory NLL, under a joint Gaussian-process scoring that accounts for temporal correlation unattained by any baseline, indicates a sharp-predictive distribution.

Result 2 - F/T prediction informs the trajectory prediction. Removing the F/T head and generating motion alone degrades trajectory RMSE $0.025 \rightarrow 0.094$ m, a larger regression than dropping any single conditioning signal (Table 2). Force and torque are the causal determinants of item displacement under the blade; therefore, a model that predicts trajectory without co-generating the F/T signature discards the dynamical mechanism and degenerates to kinematic extrapolation.

Result 3 - Deterministic regression cannot represent occluded-contact. The regression baseline predicts both modalities as point estimates and trails even the weakest ablated *BDWM* variant on both (trajectory RMSE 0.056 vs. 0.025 m; force/torque RMSE 10.2 vs. 5.75 N), with no distributional scores defined (Table 1). A single-mode predictor collapses the multimodal $(\text{scene}, \text{plan}) \rightarrow$ dynamics mapping induced by occluded contents onto its conditional mean. Taken with Results 1 and 2, this establishes the two conditions that the baselines each violate and *BDWM* jointly satisfies: contact dynamics must be modeled generatively, and force/torque and trajectory must be co-generated rather than predicted in isolation.

Result 4 - Pre-stow image conditioning and post-stow visual latent co-generation are principal drivers of dynamics fidelity. Removing the pre-stow image conditioning inflicts the worst trajectory degradation of any ablation (CRPS $4.5\times$, RMSE nearly $4\times$; Table 2): since the (often occluded) item configuration is visible only through the image, stripping it collapses the model's ability to anticipate where the blade will meet resistance. Removing the post-stow vision supervision ($\mathscr{L}_{\text{flow}}^{\text{vis}}$) doubles trajectory RMSE and raises F/T RMSE in step ($5.75 \rightarrow 9.29$ N), direct evidence that forcing the shared latent to also explain the post-sweep scene improves the physical trace it decodes, not the flow-matching head alone.

Result 5 - Action conditioning primarily governs force calibration. Dropping the planned-sweep waypoints moderately reduces point accuracy (F/T RMSE $5.75 \rightarrow 8.08$ N) but sharply miscalibrates the force distribution (F/T NLL $3.54 \rightarrow 7.16$; Table 2). As the sweep plan causally determines the contact forces, its removal produces the largest degradation in F/T calibration among all conditioning signals.

Qualitative Result 6 - Generated distribution of sample traces versus the ground truth. Figure 3 depicts the comparison between ground truth kinematic and F/T traces along with distribution of generated traces using the BDM. Also shown are the pre- and post-stow images for situating the reader in the visual physics of each bin and and its content.

5.3 Downstream Task: Predicting Created Space

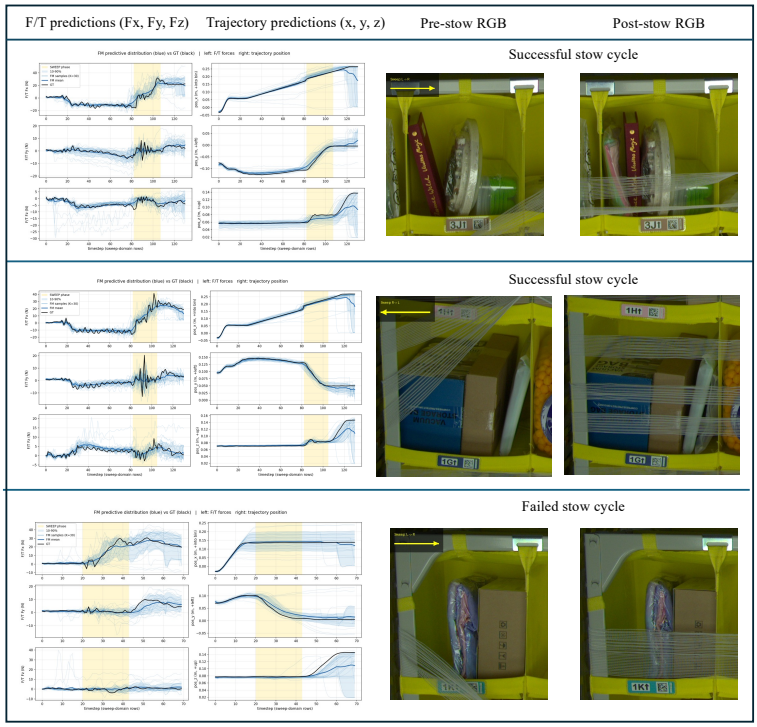

Figure 3: **Qualitative predictions of $\widehat{\text{BDWM}}$ across representative stow cycles.** Each row shows a single stow episode. Left two columns: the model's predictive distribution (blue) versus ground truth (black solid) for force/torque and end-effector trajectory. Light blue traces are K=50 posterior samples, the dark-blue line is the sample mean, and the shaded band spans the 10–90% predictive interval; the yellow region marks the SWEEP phase. Right two columns: the corresponding pre-stow and post-stow RGB observations of the bin, with the arrow indicating sweep direction. Additional results in Figure 4.

We freeze the world model and, for each cycle, draw K stochastic pre-contact rollouts from it (Appendix Figure 5). A heteroscedastic Gaussian head regresses the executed sweep length or "space created" in meters by the blade as it sweeps in the bin. Each rollout's physics latent sequence $z_{\text{phys}} \in \mathbb{R}^{200\times256}$ is encoded by a small temporal CNN; the predicted post-stow DINOv3 patch field z_{vis} is mean$\|$max-pooled over patches and projected, then concatenated per rollout. The fused per-rollout features are aggregated across the K=50 rollouts and passed through a mean head and a (bounded) log-variance head, trained with a Gaussian NLL loss so the head reports a calibrated predictive σ. We report the **MAE** and predictive **STD** (σ) against a learned regression baseline (an image/geometry regressor). On a common held-out set scored against the geometric ground-truth sweep profile, the head attains MAE 0.021 m, compared with the regression baseline MAE (0.026 m), Table 3. We ablate how the K samples are aggregated (mean, medoid, and a learned attention pool) in Section A.7. We also evaluate a second downstream task trained on the frozen *BDWM* latents: Stow Outcome Classification, shown in Section A.8 in the Appendix.

Table 3: **A frozen BDWM encodes information that drives downstream readouts.** Lightweight heads over $K = 50$ frozen-model rollouts (no gradient flow-back) outperform a task-specific regression baseline on prediction of created space.

Method	MAE (m) ↓	STD (σ) ↓
Ours	**0.021**	**0.015**
Regr. B-line	0.026	0.017

5.4 Online Forecasting: Chunking BDWM

We adapt the *BDWM* from full-trajectory generation to an online forecasting objective: given the observed prefix of a sweep $[0,t)$, predict the next n-step horizon $[t,t+n)$ of the coupled force/torque and end-effector trajectory, as shown in Figure 6. The backbone is inherited unchanged from our base model. The only modification is how the observed prefix is supplied and how the loss is applied. We encode the raw prefix signal per-timestep into a set of context tokens and append them to the

Table 4: **Conditioning on the observed sweep prefix resolves part of the pre-contact uncertainty.** BDWM-Chunk forecasts the $n = 25$ step (500,ms) horizon given an observed F/T and trajectory prefix. BDWM-Batch predicts the full trace from pre-contact inputs only. The gap quantifies the extent to which the occluded-contact uncertainty can be reduced by observing the sweep unfold. We present this as an initial study; a characterization across prefix length and horizon is left to future work.

	Force/Torque				Trajectory			
Method	CRPS ↓	GP-NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	GP-NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
BDWM-Batch	3.19	3.54	5.75	4.12	0.009	−4.78	0.025	0.005
BDWM-Chunk	**2.95**	**3.21**	**5.5**	**4.1**	**0.008**	**−4.88**	**0.022**	**0.005**

self-attention sequence $[\text{phys}; \text{vis}; \text{wp}; \text{ctx}]$, where they attend through the same sublayers as every other token. The flow denoiser still generates the full latent from noise, but the flow-matching loss is supervised only on the target-horizon tokens, while the autoencoder reconstruction remains over all valid steps to keep the decoder sharp. At train time, the split point t is sampled to yield a variable-length prefix (minimum 10 steps, 200 ms) and a fixed forecast horizon of n=25 steps (500 ms at 50 Hz). Pipeline and qualitative results of model are in Figure 6 and Figure 7 of the Appendix.

Initial Results. Table 4 compares the chunking variant (BDWM-Chunk) against the one-shot model (BDWM-Batch) under the same probabilistic protocol. Conditioning on the observed prefix improves every metric on both modalities - trajectory CRPS $0.009 \to 0.008$, RMSE $0.025 \to 0.022$ m, NLL $-4.78 \to -4.88$; force/torque CRPS $3.19 \to 2.95$, RMSE $5.75 \to 5.5$ N - consistent with the prefix resolving part of the occluded-contact uncertainty that the pre-contact model must otherwise integrate over. Anchoring the forecast to the observed timeline and re-issuing it as the sweep unfolds enables *mid-sweep* decision capabilities: the model can revise its prediction of the remaining horizon online, supplying the abort and affordance-selection signals that a one-shot pre-contact prediction cannot update once contact begins. We present this as an initial study rather than a validated capability; a rigorous characterization with forecast accuracy as a function of prefix length and horizon, and the downstream value of online re-forecasting, is left to future work.

6 Limitations and Future Work

We evaluate against logged robot executions and their realized outcomes rather than running new closed-loop trials, which keeps the results reproducible and deployment-realistic but demonstrates calibrated anticipation of how a sweep will unfold, not closed-loop failure prevention. The clearest next step is to deploy the predictions as an abort or behavior-selection signal and measure the resulting change in success rate; this requires characterizing whether the per-cycle spread thresholds into a usable failure detector. We study a single task and embodiment: generalization is probed across bins and behaviors via held-out splits, but not across robots. Since the model generates the physical consequences of an action from pre-contact inputs, it is, in principle, an offline dynamics simulator for training or evaluating policies without robot time. We view this, together with the richer downstream use of the co-generated post-stow latent, as the most promising direction opened by treating contact dynamics as a world model.

7 Conclusion

We present *BDWM*, a factored latent world model of robotic bin dynamics that jointly generates the force/torque and end-effector-trajectory traces, along with the post-contact scene latent, all from pre-contact inputs, with a single flow-matching velocity field transporting a physics and a vision latent in tandem. Our experiments establish three claims: 1) *BDWM* beats trajectory-only flow policies and a deterministic regression baseline on F/T and trajectory prediction, and removing the F/T prediction head degrades the predicted trajectory, 2) image conditioning and co-generating post-stow scene drive dynamics fidelity, and 3) frozen world-model latents drive lightweight readouts for space created and stow outcome, turning the model into pre-execution decision features. These results point to two principles beyond bin manipulation. First, flow matching is a means of modeling uncertainty in world models, not just cloning behavior. Second, force/torque should be treated as an essential prediction target wherever it is sensed and co-generating physical and visual futures is a general lever for contact-rich world models.

AI USE STATEMENT

We used generative AI tools as writing and coding assistants during the preparation of this work. Specifically, AI assistance was used to (i) improve the clarity and phrasing of prose in the manuscript; (ii) draft and format LaTeX, including tables and equations; (iii) write portions of the code, including evaluation, plotting, and analysis scripts; and (iv) summarize and cross-check descriptions of our own codebase (e.g., confirming which ablations, metrics, and loss terms are implemented) against the corresponding source. All research contributions - the problem formulation, model design, experimental protocol, and analysis - originate with the authors, and generative AI was not used to generate research ideas, design experiments, produce results, or derive claims. All AI-assisted text was read and edited by the authors; all AI-written code was reviewed, tested for correctness, and validated against expected behavior by the authors; and every reported number, metric definition, and citation was verified by the authors against the code and original sources. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have taken several steps to support reproducibility. The model architecture, latent representations, conditioning design, losses, and full training objective are specified in Section 4, with all hyperparameters and the training schedule given in Appendix. The stochastic inference procedure, including the SDE sampler, churn, guidance weight, number of Euler steps, and ensemble size K, is described in Section 5 and Appendix A.4. Our evaluation protocol - the data splits (80/10/10, seed 42), the fixed held-out cycle set, the pooling of (timestep, channel) elements, and the precise definitions of every metric are detailed in the Appendix. Additional ablations are also provided in the Appendix. The dataset used in this work will be publicly released. The code cannot be released; we therefore provide complete descriptions of the architecture, training, data channels, and preprocessing to enable reproduction.

REFERENCES

Md Hazrat Ali, K Aizat, K Yerkhan, T Zhandos, and O Anuar. Vision-based robot manipulator for industrial applications. *Procedia computer science*, 133:205–212, 2018.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Sang Hun Cheong, Brian Y Cho, Jinhwi Lee, ChangHwan Kim, and Changjoo Nam. Where to relocate?: Object rearrangement inside cluttered and confined environments for robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7791–7797. IEEE, 2020.

Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2086–2092. IEEE, 2023.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL `https://arxiv.org/abs/1912.01603`.

Haotian He, Zeyu Yan, Qipeng Liu, Ning Guo, and Wenzhao Lian. Fawam: Force-aware world action models for closed-loop contact-rich manipulation. *arXiv preprint arXiv:2606.08555*, 2026.

Nicolas Hudson, Josh Hooks, Rahul Warrier, Curt Salisbury, Ross Hartley, Kislay Kumar, Bhavana Chandrashekhar, Paul Birkmeyer, Bosch Tang, Matt Frost, Shantanu Thakar, Tony Piaskowy, Petter Nilsson, Josh Petersen, Neel Doshi, Alan Slatter, Ankit Bhatia, Cassie Meeker, Yuechuan Xue, Dylan Cox, Alex Kyriazis, Bai Lou, Nadeem Hasan, Asif Rana, Nikhil Chacko, Ruinian Xu, Siamak Faal, Esi Seraj, Mudit Agrawal, Kevin Jamieson, Alessio Bisagni, Valerie Samzun,

Christine Fuller, Alex Keklak, Alex Frenkel, Lillian Ratliff, and Aaron Parness. Stow: Robotic packing of items into fabric pods, 2025. URL https://arxiv.org/abs/2505.04572.

Simon Lang, Mihai Alexe, Mariana CA Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D Dueben, Sara Hahner, et al. Aifs-crps: ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(1):18, 2026.

Yang Li, Zhaxizhuoma Zhaxizhuoma, Hongru Jiang, Junjie Xia, Hongquan Zhang, Jinda Du, Yunsong Zhou, Jia Zeng, Ce Hao, Jieji Ren, et al. Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8911–8920, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11058–11068. IEEE, 2025.

Chenrui Ma. Transition flow matching, 2026. URL https://arxiv.org/abs/2603.15689.

Manthan Patel, Jonas Frey, Mayank Mittal, Fan Yang, Alexander Hansson, Amir Bar, Cesar Cadena, and Marco Hutter. Defm: Learning foundation representations from depth for robotics. *arXiv preprint arXiv:2601.18923*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Max Schwarz, Christian Lenz, Germán Martín García, Seongyong Koo, Arul Selvam Periyasamy, Michael Schreiber, and Sven Behnke. Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3347–3354. IEEE, 2018.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026a.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation, 2026b. URL https://arxiv.org/abs/2602.22010.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model, 2026. URL https://arxiv.org/abs/2602.10098.

Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems*, 37:56166–56189, 2024.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, Cewu Lu, and Wenqiang Zhang. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2505.22159.

Ruiteng Zhao, Wenshuo Wang, Yicheng Ma, Xiaocong Li, Francis EH Tay, Marcelo H Ang Jr, and Haiyue Zhu. Fd-vla: Force-distilled vision-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2602.02142*, 2026.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling, 2025. *URL https://arxiv. org/abs/2505.15659*.

A Appendix

A.1 Data Channels Used

Table 5: Force/torque (F/T) trace channels ($C_{\text{ft}} = 13$). The 7-channel *blade+gripper* variant retains indices 0–5 and 12, dropping the six raw wrist channels (6–11).

Idx	Channel	Unit	Signal
0–2	blade_{fx,fy,fz}	N	Blade-tip contact *force* (15 Hz), inertia-compensated: blade weight/acceleration subtracted, leaving the pure contact wrench against bin contents and walls.
3–5	blade_{tx,ty,tz}	N m	Blade-tip contact *torque*, same sensor and compensation.
6–8	wrist_f_{fx,fy,fz}	N	Wrist *force* raw, 10 Hz, raw: no inertia compensation exists for this sensor, so contact is mixed with the arm's own inertial and gravity loads; heavy-tailed and varying.
9–11	wrist_t_{tx,ty,tz}	N m	Wrist *torque*, same caveats.
12	gripper_ft_fz	N	Gripper force, scalar magnitude (frame-independent): load felt while holding/releasing the item.

Table 6: Trajectory channels ($C_{\text{traj}} = 17$), describing the observed blade-tip motion and auxiliary actuator states. Together with the F/T trace this forms the 30-D (or 24-D for the 7-channel F/T variant) motion target predicted jointly by the model. The separate 13-D *action* conditioning input uses the same pose/velocity layout (indices 0–12) for the *commanded* blade-tip motion, without the extras.

Idx	Channel	Unit	Signal
0–2	pos_{x,y,z}	m	Blade-tip position in the bin frame; x is the push (sweep) direction.
3–6	quat_{w,x,y,z}	–	Blade-tip orientation (unit quaternion).
7–9	linvel_{x,y,z}	m/s	Blade-tip linear velocity.
10–12	angvel_{x,y,z}	rad/s	Blade-tip angular velocity.
13	blade_position	m	Blade extension along its own actuator (deployment depth).
14	gripper_position	m	Gripper opening.
15–16	pedestal_pos_{0,1}	m	Pedestal actuator positions (workcell platform posture).

A.2 Peak-Weighted Loss

$$\mathscr{L}\text{peak} = \frac{1}{\sum_i m_i w_i} \sum_i m_i, w_i \left| \text{decoder ft}(\hat{z})_i - \tau_i^{\text{GT}} \right|^2, \qquad w_i = 1 + \alpha, \frac{|\tau_i^{\text{GT}}|}{\text{scale}}, \;\; \alpha = 4, \tag{5}$$

where scale is the per-channel standard deviation of τ^{GT}.

A.3 Spatio-Temporal Grounding

This term couples *what* the model attends to in the pre-stow scene to *when* and *how hard* the blade makes contact. A grounding loss $\mathscr{L}_{\text{ground}}$ supervises the physics→RGB cross-attention of the last block, $A \in \Delta^N$ per physics token (each row an attention distribution over the N image patches). For physics token i, let $p_i = (Y, Z)$ be the commanded blade position at that step and $\{c_n\}$ the patches' bin-frame coordinates. We form a soft spatial target concentrated near the blade,

$$t_{i,n} = \mathrm{softmax}_n\left(-\frac{\|c_n - p_i\|^2}{2\sigma^2}\right), \quad \sigma = 0.05 \text{ m}, \tag{6}$$

masking patches with no valid depth (distance set to ∞). We push the attention toward this target with a per-token cross-entropy $\mathrm{CE}_i = -\sum_n t_{i,n} \log A_{i,n}$, weighted by the normalized blade contact-force magnitude,

$$w_i = \mathrm{clip}\left(\frac{\|f_i\|}{\max_j \|f_j\|}, f_{\min}, 1\right) \cdot \mathbb{1}[\mathrm{valid}_i], \quad f_{\min} = 0.1, \tag{7}$$

so that high-force contact timesteps dominate and free-space timesteps contribute little: $\mathscr{L}_{\text{ground}} = \left(\sum_i w_i \mathrm{CE}_i\right) / \sum_i w_i$. This ties the spatial pre-contact representation to the temporal dynamics the model must generate.

A.4 Training Details

BDWM uses $D = 18$ blocks, 8 heads, and hidden width 512, with every attention/MLP sublayer zero-gate-initialized so each block is an identity map at initialization. We train for up to 300 epochs (early stopping around 150) at batch size 24 across 8 GPUs, with AdamW (learning rate 10^{-4}, weight decay 10^{-5}, 1000-step warmup), gradient clipping at 1.0, and an exponential moving average (EMA) of the weights (decay 0.9999) used for all evaluation.

Classifier-free guidance. During training we apply joint conditioning dropout with probability 0.1: on the drop event the RGB, depth, segmentation, and waypoint-validity signals are zeroed (the already-null AdaLN embedding is left in the graph). This stops the flow from memorizing a deterministic map (Yao et al., 2024; Ma, 2026). *BDWM* training details in Section A.4.

A.5 Metrics

Notation. For a fixed channel, we pool all valid (timestep, cycle) pairs into points $n = 1, \ldots, N$. Each point carries K i.i.d. world-model samples $x_n^{(1)}, \ldots, x_n^{(K)}$, with ensemble mean $\mu_n = \frac{1}{K}\sum_k x_n^{(k)}$ and empirical standard deviation $\sigma_n = \left(\frac{1}{K}\sum_k (x_n^{(k)} - \mu_n)^2\right)^{1/2}$, and a ground-truth value y_n. Scores are averaged over channels.

CRPS (continuous ranked probability score). A proper score that rewards accurate, appropriately-dispersed forecasts (lower is better), penalizing both over- and under-dispersion. We use the ensemble (energy) estimator:

$$\mathrm{CRPS}_n^{\mathrm{ens}} = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\left|x_n^{(k)} - y_n\right|}_{\text{accuracy: samples vs. truth}} - \underbrace{\frac{1}{2K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\left|x_n^{(i)} - x_n^{(j)}\right|}_{\text{sharpness: within-ensemble spread}}, \tag{8}$$

where $x_n^{(k)}$ is the k-th sample and y_n the ground truth at point n. The first term shrinks as samples concentrate near y_n; the second rewards spread, so the score is minimized by an ensemble that is both accurate and calibrated. The reported CRPS is $\frac{1}{N}\sum_n \mathrm{CRPS}_n^{\mathrm{ens}}$.

Table 7: Probabilistic trace prediction on 24,783 held-out cycles (K=50, SDE sampler). Reporting NLL instead of GP-NLL. Deterministic models produce a single trace. Their distributional scores are marked "dnmu" - does not model uncertainty. π_0, $\pi_{0.5}$ and ForceVLA do not model force (F/T entries marked "dnm" - does not model force).

	Force/Torque				Trajectory			
Method	CRPS ↓	NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
Regression baseline	dnmu	dnmu	10.213	dnmu	dnmu	dnmu	0.0561	dnmu
π_0 (Black et al., 2024)	dnm	dnm	dnm	dnm	0.037	13.14	0.068	0.027
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	dnm	dnm	dnm	dnm	0.036	14.55	0.067	0.026
ForceVLA (Yu et al., 2025)	dnm	dnm	dnm	dnm	dnmu	dnmu	0.034	dnmu
Ours	**3.19**	**3.33**	**5.75**	**4.12**	**0.009**	**−4.91**	**0.025**	**0.005**

NLL (per-element Gaussian). We model each point as $\mathcal{N}(\mu_n, \sigma_n^2)$ - a Gaussian with the ensemble mean μ_n and empirical variance σ_n^2 - and score the truth y_n:

$$\mathrm{NLL} = \frac{1}{N}\sum_{n=1}^{N}\left[\tfrac{1}{2}\log\big(2\pi\sigma_n^2\big) + \frac{(y_n-\mu_n)^2}{2\sigma_n^2}\right]. \tag{9}$$

Lower is better; the score penalizes both large residuals $(y_n-\mu_n)^2$ and miscalibrated predictive variance σ_n^2 (over-confident predictions are penalized by the second term, over-dispersed ones by the first).

RMSE and best-of-K. RMSE of the ensemble mean measures point accuracy:

$$\mathrm{RMSE} = \sqrt{\tfrac{1}{N}\sum_{n=1}^{N}(\mu_n - y_n)^2}. \tag{10}$$

The best-of-K (minRMSE) error is an oracle-selection measure of whether any sampled rollout matches the realized dynamics: with per-cycle rollouts $\hat{y}_{t,c}^{(k)}$ over timesteps $t = 1,\ldots,T$ and channel c,

$$\mathrm{minRMSE}_c = \min_k \sqrt{\tfrac{1}{T}\sum_{t=1}^{T}\big(\hat{y}_{t,c}^{(k)} - y_{t,c}\big)^2}. \tag{11}$$

It rewards a model whose predictive distribution covers the true outcome even when the mean does not, complementing RMSE (which scores the mean) and CRPS (which scores the full distribution).

A.6 Data Splits and Inference Protocol

Data and splits. We evaluate on logged real-robot blade-sweep cycles with synchronized pre-/post-stow RGB-D and segmentation masks, multi-sensor force/torque traces, commanded and observed trajectories, and outcome labels. The dataset contains 247,824 trainable cycles, split $80/10/10$ into train/validation/test (seed 42).
Inference protocol. Unless noted, we draw $K = 50$ samples per cycle from the frozen world model with the SDE sampler (churn 0.5, 150 Euler steps, guidance $g_{\mathrm{cfg}} = 1$). All numbers below are on held-out 24,783 cycles and use the EMA weights of the checkpoint.

Table 8: **Spatio-temporal RoPE is essential.** Removing the shared-bin-frame rotary positional encoding (No RoPE) degrades every metric on both modalities, roughly doubling trajectory RMSE (0.025 → 0.052 m) and collapsing calibration (trajectory NLL −4.78 → 5.23), confirming that anchoring image and physics tokens to a common metric frame is what lets cross-attention align where and when the blade makes contact.

	Force/Torque				Trajectory			
Method	CRPS ↓	NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
No RoPE	4.98	7.23	7.89	6.25	0.039	5.23	0.052	0.047
Ours	**3.19**	**3.54**	**5.75**	**4.12**	**0.009**	**−4.78**	**0.025**	**0.005**

Table 9: **The spatio-temporal grounding loss improves the predicted trace.** Dropping $\mathscr{L}_{\text{ground}}$, which aligns physics-token attention with the pre-stow patches the blade contacts, worsens every metric, most notably trajectory CRPS (0.009 → 0.027) and NLL (−4.78 → −3.84), showing that tying the spatial pre-contact representation to the temporal contact dynamics sharpens and calibrates the generated trace.

	Force/Torque				Trajectory			
Method	CRPS ↓	NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
No $\mathscr{L}_{ground}$	3.95	3.80	5.95	5.89	0.027	−3.84	0.032	0.019
Ours	**3.19**	**3.54**	**5.75**	**4.12**	**0.009**	**−4.78**	**0.025**	**0.005**

Table 10: **The SDE sampler is better-calibrated than the ODE sampler.** Under identical checkpoints and K=50 rollouts, stochastic (SDE, churn 0.5) sampling improves every metric over the deterministic ODE sampler, most clearly in calibration (trajectory NLL −3.78 → −4.78; force/torque NLL 3.90 → 3.54), so we use the SDE sampler throughout. Lower-is-better except as noted; NLL in nats/timestep, force/torque in N, trajectory in m.

	Force/Torque				Trajectory			
Method	CRPS ↓	NLL ↓	RMSE (N) ↓	best-of-K (N) ↓	CRPS ↓	NLL ↓	RMSE (m) ↓	best-of-K (m) ↓
ODE Inference Sampler	3.20	3.90	6.46	4.43	0.011	−3.78	0.029	0.008
SDE Inference Sampler	**3.19**	**3.54**	**5.75**	**4.12**	**0.009**	**−4.78**	**0.025**	**0.005**

A.7 Methods of Sampling K

3 ways to get K:

1. Let the model learn it: feed all K, and then pool it, pass it to the model. Model learns the best aggregation. 2. Mean of K: Choose K (50), take the mean of it and then feed it to the model. 3. Medoid method: Selecting a central subset, top x most central of K.

The GT for a cycle is a single observed realization, so if the true distribution is bimodal, the "right" target is ambiguous - the medoid/dominant-mode sweep is the typical outcome, which is usually what you want at inference, but it won't always match that cycle's recorded label. That distinction determines whether the fix is "best-of-K select then pool" vs "feed all K and let the head learn the reduction."

Table 11: Sample-aggregation ablation. Pooling the K=50 rollouts by learned attention beats mean- and medoid-pooling.

Aggregator	NLL ↓	RMSE (m) ↓
Mean	−3.10	0.0182
Medoid	−3.21	0.0173
Learned Attention	**−3.25**	**0.0170**

A.8 Stow-Outcome Classification

A head classifies each cycle as {Success, Failure} (class balance ≈ 85/15%). For each of the K=50 rollouts we assemble: (i) the pooled physics latent z_{phys} (same CNN+masked mean‖max trunk as above), (ii) the pooled post-stow-vision latent z_{vis}, (iii) the new item's (to be inserted) dimensions, (iv) the affordance insert pose of the new item, and (v) the predicted sweep length from our dynamics model. A learned two-layer scorer produces a softmax attention weight over the K rollouts from their motion latents; this single weight vector then pools every modality, so vision, physics, affordances are aggregated consistently with the dynamics. The pooled features are fused and classified into the two outcomes, trained with inverse-frequency-weighted cross-entropy with best-balanced-accuracy checkpointing. Results in table 12.

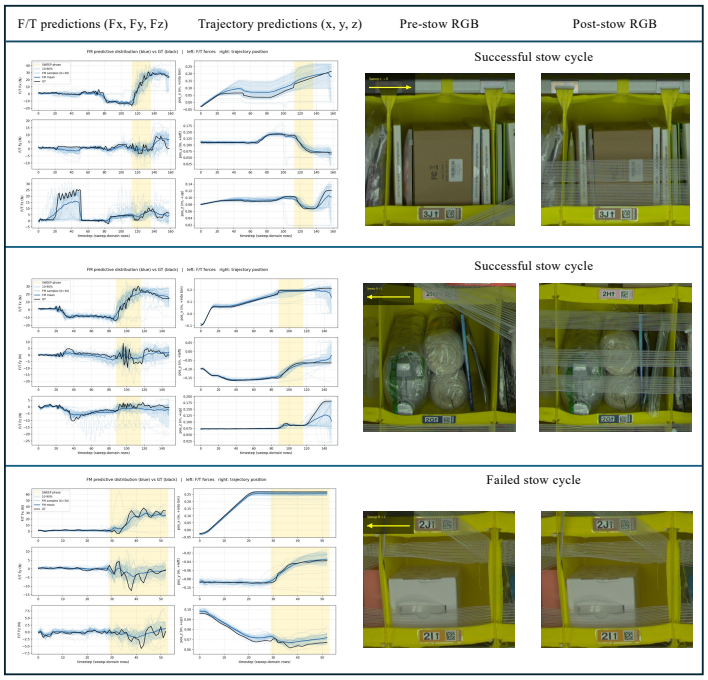

Figure 4: **Additional qualitative predictions on successful and failed stow cycles.** Columns and color conventions as in Figure 3; each row is one stow episode.

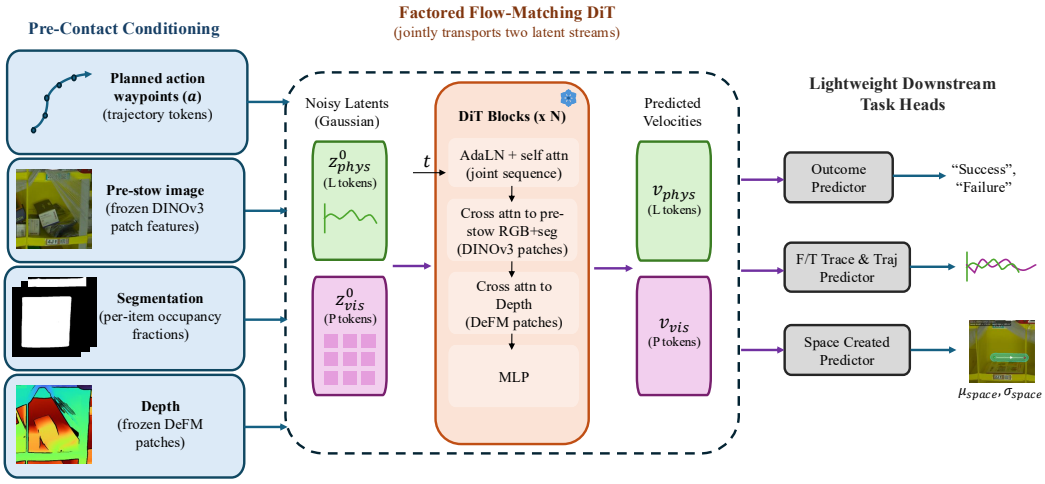

Figure 5: **Downstream-readout pipeline over a frozen BDWM.** For each cycle we draw K=50 stochastic pre-contact rollouts from the frozen world model (no gradients flow back). Each rollout's physics latent z_{phys} is encoded by a small temporal CNN and pooled over time, and the predicted post-stow vision latent z_{vis} is pooled over patches and projected, forming a per-rollout feature. *Space-Created Prediction:* the K features are aggregated and passed through a heteroscedastic Gaussian head (mean and bounded log-variance) for a calibrated sweep-length estimate. *Stow-Outcome Classification:* each feature is augmented with the new item's dimensions, its affordance insert pose, and the predicted sweep length; a learned attention weight over the K rollouts pools every modality before a fused classifier predicts {Success, Failure}.

Table 12: **Stow-Outcome Prediction (Success vs. Failure).** Lightweight heads over K=50 frozen-model rollouts (no gradient flow-back).

Stow Outcome Classification		
Method	Prob. Success	Prob. Failure
Ours	0.90	0.10
Ground Truth	0.85	0.15

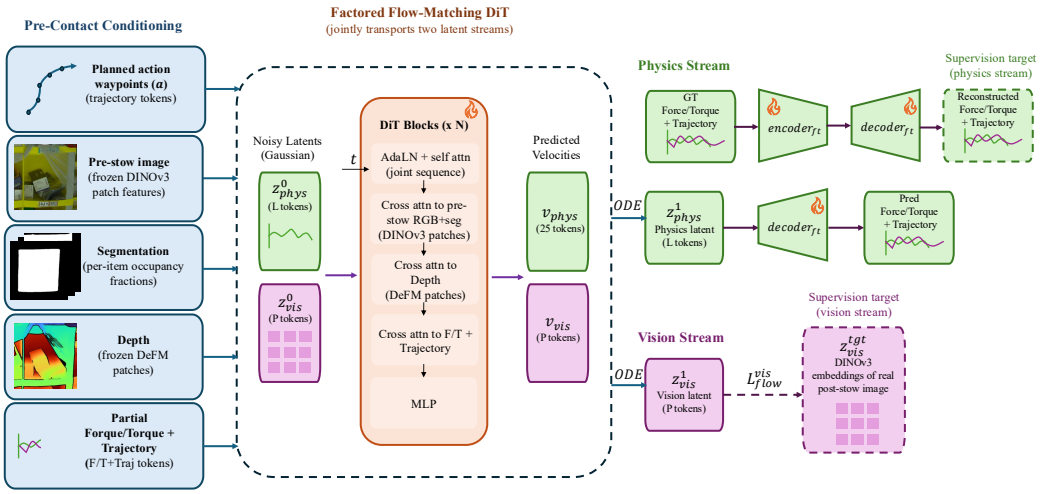

Figure 6: **Online-forecasting (chunking) pipeline.** The base BDWM backbone is reused unchanged; only the input and loss masking differ. The observed sweep prefix is encoded per-timestep into context tokens appended to the self-attention sequence $[\text{phys}; \text{vis}; \text{wp}; \text{ctx}]$, RoPE-aligned to the physical-state tokens and gated by a validity mask so no horizon step ever enters as input. The flow denoiser generates the full latent from noise, but the flow-matching loss is supervised only on the target-horizon tokens, while the autoencoder reconstruction remains over all valid steps. At inference the model re-forecasts the remaining horizon as the sweep unfolds, enabling mid-sweep abort and affordance-selection signals.

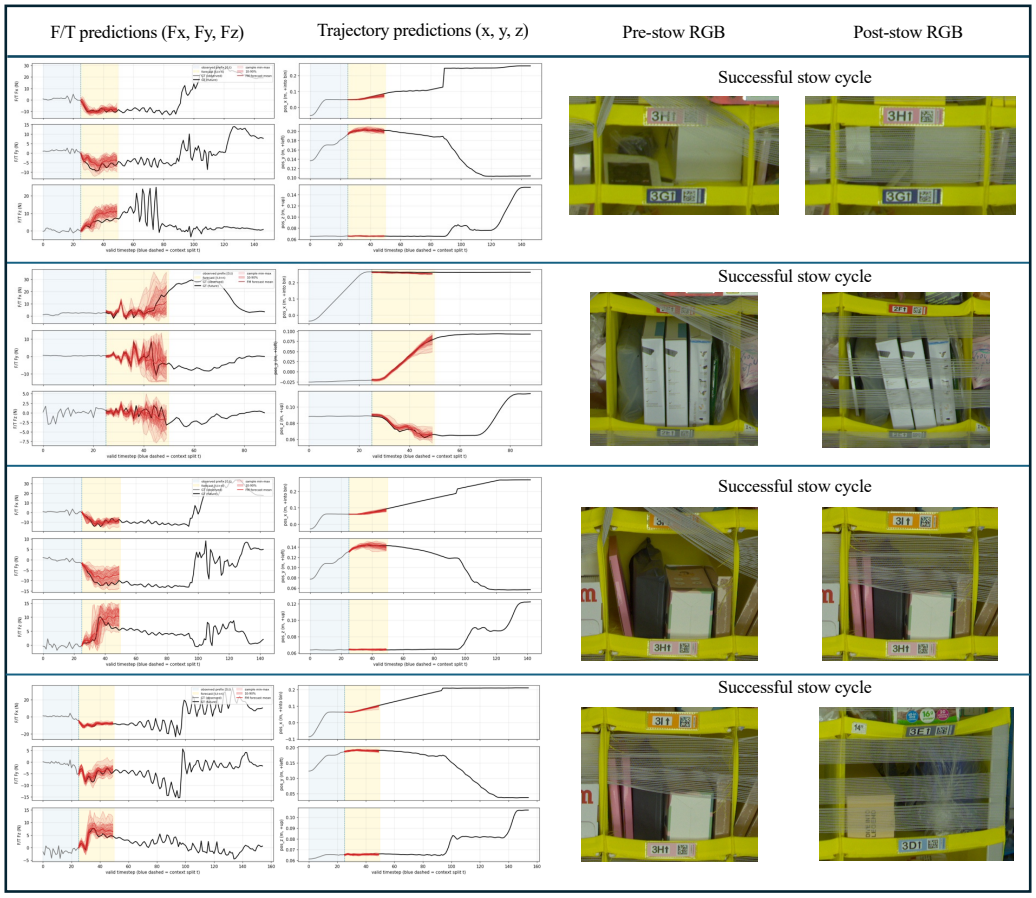

Figure 7: **Qualitative online-forecasting (chunking) predictions.** For representative sweep cycles, BDWM-Chunk observes a prefix $[0,t)$ of the sweep (grey) and forecasts the next n=25-step (500 ms) horizon $[t,t+n)$ of the coupled force/torque and end-effector trajectory. Black is ground truth, the dark-red line the K=50-sample mean, and the shaded band the 10–90% predictive interval; the vertical marker separates the observed prefix from the forecast horizon. Conditioning on the observed prefix tightens the predictive band over the pre-contact-only model, consistent with the prefix resolving part of the occluded-contact uncertainty.