FORCEWAM: DECOUPLING FUTURE AND RESPONSE LEARNING FROM DEMO AND PLAY DATA FOR CONTACT-RICH MANIPULATION

Anonymous authors
Paper under double-blind review

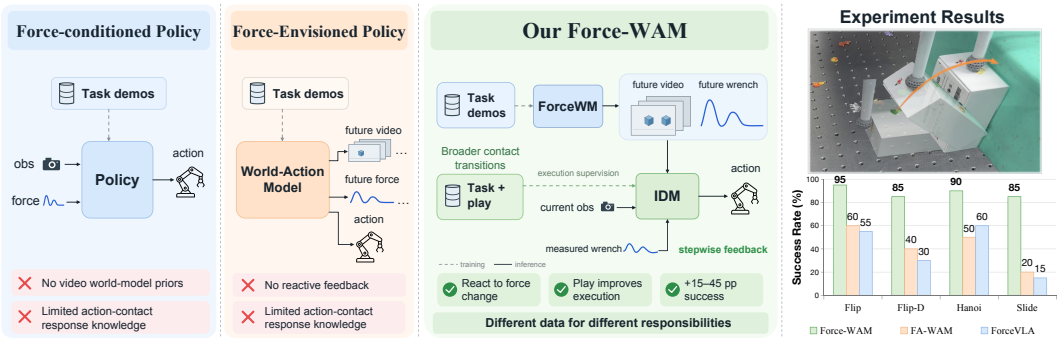

Figure 1: **Motivation of ForceWAM.** The illustrated end-to-end force-conditioned and force-envisioned policies use only task demonstration data: the former lack pretrained video-world-model priors, while the latter lack within-chunk force feedback. ForceWAM combines the pretrained prior with task demonstrations for task-compatible visual-wrench futures and action-labeled play for IDM execution learning, achieving the highest mean success across all four real-robot conditions.

ABSTRACT

Expert demonstrations show how to complete contact-rich tasks, but provide limited coverage of how interaction forces respond to different actions. Task-agnostic play provides broader experience of contact dynamics without specifying task goals. To fully leverage these two types of data, we introduce ForceWAM, a force world action model that decouples knowledge of task-compatible interaction outcomes from knowledge of how local actions change contact. A world model learns from successful demonstrations to jointly predict future video and six-axis force/torque trajectories. Together, these futures describe the expected evolution of motion and contact during the task. A separate inverse dynamics model (IDM) learns from both task and play, mapping current observations and recorded visual-wrench futures to their associated robot actions. Play data thus provide knowledge of force responses for goal-directed execution. At deployment, the IDM reactively translates generated futures into actions, reusing them with fresh wrench feedback between world-model updates. Across real-robot force-regulated sliding, box flipping with and without disturbances, and ring placement, ForceWAM achieves 85–95% mean success, surpassing world-modeling and reactive baselines. Our experiments further show that adding task-agnostic play data improves success by 15–45 percentage points, demonstrating its value for learning contact-rich execution.

1 INTRODUCTION

Robotic manipulation is often cast as predicting actions that reach a desired visual state. Yet contact-rich tasks are decided not only by where the robot moves, but by how interaction forces evolve along the way. Two visually similar configurations can correspond to insufficient contact, damaging force, or a jam. Classical force control therefore places interaction feedback at the center of execution (Hogan,

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.

1985), while recent learned systems show that force and touch improve contact-rich manipulation (Liu et al., 2025; Xue et al., 2025; Yu et al., 2025; Chen et al., 2026; Wang et al., 2026; Zhuo et al., 2026). We call this learning problem *force-centric manipulation*: a policy must not merely sense force, but learn the physical regularities needed to regulate it. Recent methods learn adaptive compliance or hybrid force-position control (Hou et al., 2025; Fang et al., 2026; Li et al., 2026c). We aim to learn the full force-centric manipulation task directly from data. However, the main obstacle is a mismatch between where these regularities appear in data and how policies learn them.

Expert demonstrations, which we call *task data*, show successful task execution. A box-pushing demonstration at roughly $10\,\mathrm{N}$, for example, need not show how force changes when the robot presses farther, backs away, or encounters a disturbance. By contrast, *task-agnostic play data* record exploratory pressing, sliding, and release, providing additional visual-wrench transitions and associated robot actions. Play transitions need not demonstrate successful behavior for a given task instruction, but they still provide recorded future-action pairs for inverse-dynamics learning. End-to-end force-aware learners (Yu et al., 2025; Chen et al., 2026) still couple two questions in one task-supervised mapping: *what* transition advances the task, and *which action* realizes that transition. Our question is: can we use these two data sources more effectively to learn the distinct knowledge each provides?

Video planning with inverse dynamics provides a foundation for learning actions from interaction outcomes: recorded transitions supervise the inverse model, while generated futures guide execution (Du et al., 2023; Li et al., 2026b). For contact-rich manipulation, a visual future can leave the desired evolution of contact forces unspecified. Force- and tactile-aware models have demonstrated the value of predicting contact signals and using them for reactive control (He et al., 2026; Ma et al., 2026). We study visual-wrench futures as the condition for learning execution from task and play, associating actions with both the motion and the contact evolution they produce.

We introduce **ForceWAM** (Figure 1), which uses visual-wrench futures to decouple knowledge of task-compatible interaction outcomes from knowledge of how local actions change contact. Successful task demonstrations train the force world model (ForceWM) to jointly generate future video and six-axis wrench trajectories. The inverse dynamics model (IDM) learns from task and play segments to predict the robot actions that realize a visual-wrench future. During training, it conditions on the current observation and the future recorded in each segment. Any segment that records actions and their resulting motion and contact outcomes can supervise this action prediction, even if it does not complete a task. At deployment, ForceWM-generated futures replace the recorded future conditions used in IDM training, specifying the interaction that execution should realize. The IDM reuses each generated reference within an action chunk, reactively updating its predictions with newly measured wrench feedback.

We evaluate ForceWAM on force-regulated sliding, box flipping with and without disturbances, and ring placement, achieving 85–95% mean success across the four evaluation conditions. Our experiments assess the benefit of added play and the role of visual-wrench future conditioning in execution. Our contributions are:

- We formulate contact execution as inverse dynamics conditioned on visual-wrench futures. This formulation allows different types of interaction data to supervise action prediction.
- We propose ForceWAM, whose visual-wrench future interface decouples knowledge of task-compatible interaction outcomes from knowledge of local action-contact responses. ForceWM learns from successful demonstrations. The IDM learns from task and play, mapping generated futures to actions with causal wrench feedback between refreshes.
- Across four real-robot evaluation conditions, we show that added play improves success by 15–45 percentage points with task demonstrations, ForceWM, and IDM optimizer updates held fixed. Future-conditioning ablations further show the benefit of wrench futures alongside visual futures, with measured wrench feedback retained.

2 Related Work

Force-aware manipulation. Force and tactile feedback provide contact information that is difficult to infer from vision alone (Lee et al., 2019; Suresh et al., 2024; Ai et al., 2024). ForceMimic couples force-motion demonstration capture with hybrid imitation, while ForceVLA integrates force into

vision-language-action policies and ForceVLA2 extends this line to hybrid force-position control (Liu et al., 2025; Yu et al., 2025; Li et al., 2026c). Beyond incorporating force as an observation, reactive policies use newly measured contact signals to update actions during execution (Calandra et al., 2018). RDP combines a slow latent policy with a fast tactile executor, whereas ImplicitRDP realizes within-chunk force feedback through causal attention in a single diffusion network (Xue et al., 2025; Chen et al., 2026). FA-RDP adapts inference frequency to manipulation phase, and LIFT combines reactive force injection with online corrective supervision (Zhuo et al., 2026; Wang et al., 2026). ForceWAM decouples knowledge of task-compatible interaction outcomes from knowledge of how local actions change contact: ForceWM models what a successful interaction should look and feel like, while an inverse dynamics model (IDM) turns that future into robot actions and reactively updates them with within-chunk wrench feedback. The IDM is conditioned on the current observation and visual-wrench future, which lets it learn from both task and play transitions.

World action models. World action models connect future observation prediction with robot action generation (Bi et al., 2026a;b; Cai et al., 2026), as exemplified by LingBot-VA, Cosmos Policy, DreamZero, FACT, and GE-Act 2.0 (Li et al., 2026a; Kim et al., 2026; Ye et al., 2026; Peng et al., 2026; AgiBot Research Team et al., 2026). A related route separates video planning from action inference (Xu et al., 2025): UniPi translates generated visual plans with a separately trained IDM, and VERA develops a Jacobian-based IDM that can learn independently from self-play (Du et al., 2023; Li et al., 2026b). Contact-aware models extend prediction to force and tactile dynamics for planning and reactive control (Higuera et al., 2026; Zheng et al., 2026; Zhou et al., 2026; Madan et al., 2026; NeoteAI Team & Fudan TEAI Team, 2026). Among these, FAWAM jointly predicts actions and wrenches and trains residual corrections from human interventions and policy rollouts (He et al., 2026). TacPAC uses live tactile observations and cached tactile-action predictions to correct an existing action chunk, with its corrector supervised against demonstrated actions (Ma et al., 2026). ForceWAM separates learning task-compatible outcomes from learning local action-contact responses while retaining a force-aware reactive inference process. Successful demonstrations train a visual-wrench future generator, while task and play windows directly train an IDM on their recorded futures and associated actions. At deployment, the IDM uses the predicted visual-wrench future as a task-conditioned reference and reactively corrects each action with newly measured wrench feedback within the cached future window. Thus, the two modules are decoupled in what they learn but connected through force feedback during execution, rather than using a visual future only as a static planning target. This makes task-agnostic contact transitions a direct source of supervision for executing task-conditioned visual-wrench futures.

3 METHOD

3.1 OVERVIEW

ForceWAM separates two questions that are entangled in an end-to-end policy: *what should a successful interaction look and feel like?*, and *what robot actions can realize that interaction?* Let $\mathbf{o}_t = (\mathbf{I}_t, \mathbf{w}_t)$ denote the current visual-wrench observation, let $\mathbf{g}_t$ denote a future visual-wrench reference, and let $\mathbf{A}_t = (\mathbf{a}_{t,0}, \ldots, \mathbf{a}_{t,H_a-1})$ denote a short action sequence. Our architecture factorizes their joint distribution as

$$\begin{aligned} p(\mathbf{g}_t, \mathbf{A}_t \mid \mathbf{o}_t, \ell) &= p(\mathbf{g}_t \mid \mathbf{o}_t, \ell)\, p(\mathbf{A}_t \mid \mathbf{o}_t, \mathbf{g}_t, \ell) \\ &\approx p_\phi(\mathbf{g}_t \mid \mathbf{o}_t, \ell)\, p_\theta(\mathbf{A}_t \mid \mathbf{o}_t, \mathbf{g}_t), \end{aligned} \tag{1}$$

where the first factor is ForceWM and the second is the inverse dynamics model (IDM). The IDM predicts actions from the current observation and a visual-wrench future, without a separate task instruction ℓ. At deployment, ForceWM supplies the task-conditioned future $\mathbf{g}_t$, and the IDM reactively updates actions using newly measured wrench feedback. This conditioning lets task and play data share the same inverse-dynamics objective. We train p_ϕ on successful task trajectories $\mathcal{D}_{\text{task}}$, whereas p_θ can learn from $\mathcal{D}_{\text{task}} \cup \mathcal{D}_{\text{play}}$.

At time t, let $\mathbf{I}_t$ denote the wrist image, $\mathbf{w}_t \in \mathbb{R}^6$ the measured TCP force and torque, and ℓ the task instruction. We use two trajectory sources. Successful *task data* show futures compatible with the task. *Play data* contain the same synchronized image, wrench, TCP-pose, and gripper-state streams, but need not solve or even attempt the downstream task. We therefore use play to learn local execution responses.

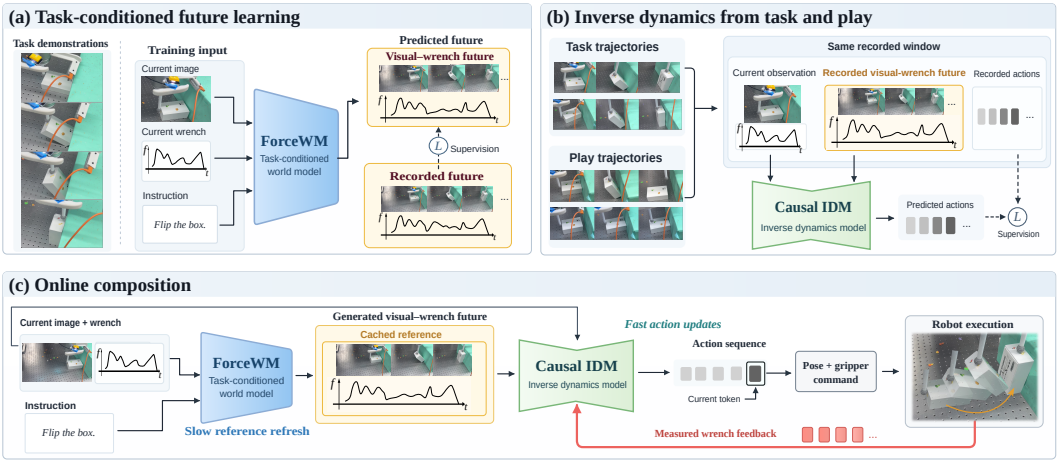

Figure 2: **ForceWAM training and deployment.** Task demonstrations train the visual-wrench future generator, while task and play data train the IDM from recorded futures and associated actions. During deployment, ForceWM supplies future conditions and the causal IDM updates actions using measured wrench feedback.

3.2 Prediction from Task Data

For a trajectory window starting at time t, we construct the future prediction target $\mathbf{g}_t = (\mathbf{z}_t, \mathbf{W}_t)$, $\mathbf{W}_t = (\bar{\mathbf{w}}_{t+1}, \ldots, \bar{\mathbf{w}}_{t+H_g})$, where $\mathbf{z}_t$ encodes future wrist images and $\mathbf{W}_t$ is a normalized future wrench trajectory. Thus, $\mathbf{g}_t$ is the target visual and contact trajectory for ForceWM. The corresponding robot actions are modeled separately by the IDM. Appendix B specifies frame offsets and encodings.

ForceWM models $p_\phi(\mathbf{g}_t \mid \mathbf{I}_t, \mathbf{w}_t, \ell)$ from successful task demonstrations. Its backbone is a pretrained spatiotemporal video diffusion model. We arrange its latent input as five semantic slots: a blank prefix, the current wrench, the current image, the future wrench trajectory, and the future visual latent. The first three slots are clamped throughout denoising, while the last two are jointly generated. A fixed packing adapter repeats low-dimensional wrench values over the spatial extent of a video-latent slot. This construction reuses one spatiotemporal backbone for the two future modalities while retaining separate visual and wrench outputs.

For a clean task future $\mathbf{g}_t$, we sample a noise level σ and train an Elucidated Diffusion Model (EDM) denoiser (Karras et al., 2022) to recover the joint future,

$$\mathcal{L}_{\mathrm{WM}} = \mathbb{E}\left[\lambda(\sigma)\, \|\mathbf{M}_{\mathrm{gen}} \odot (\mathbf{x}_t - D_\phi(\mathbf{x}_t^\sigma; \ell, \sigma))\|_2^2\right]. \tag{2}$$

Here $\mathbf{x}_t$ is the five-slot clean latent, $\mathbf{x}_t^\sigma$ adds Gaussian noise only to its generated slots, and $\mathbf{M}_{\mathrm{gen}}$ selects the future-wrench and future-visual slots. The image and wrench conditions enter D_ϕ through their clamped slots, whereas ℓ enters through text conditioning. Consequently, the blank, current-wrench, and current-image slots remain clean and receive no reconstruction loss. EDM training samples σ from a hybrid log-normal and high-noise uniform distribution. The exact mixture is given in Appendix D. At inference, ForceWM starts the two future slots from noise and samples $\widehat{\mathbf{g}}_t = (\widehat{\mathbf{z}}_t, \widehat{\mathbf{W}}_t)$. Training on successful demonstrations targets a distribution of task-compatible interactions rather than a single force trace. Multiple visual-force futures can accomplish the task.

3.3 Inverse Dynamics Learning from Task and Play

The IDM models a conditional distribution over a short sequence $\mathbf{A}_t = (\mathbf{a}_{t,0}, \ldots, \mathbf{a}_{t,H_a-1})$. The executable part of each action contains a Cartesian TCP-pose target and a gripper command. As in ImplicitRDP (Chen et al., 2026), training also predicts force-derived auxiliary targets that are discarded at execution. These targets are held fixed across all IDM comparisons. TCP poses are represented relative to the pose at the beginning of the window. The exact target construction and deterministic conversion to robot commands are given in Appendix C.

Each IDM training window contains the current image $\mathbf{I}_t$, the recorded future $\mathbf{g}_t$ constructed from later frames and wrenches in that same window, the action-rate wrench sequence, and the associated action sequence $\mathbf{A}_t$. ForceWM is not run to create IDM training inputs. The same construction applies to task and play trajectories: although a play window need not accomplish the downstream task, it still provides one feasible action sequence associated with its realized visual-wrench future.

The conditional diffusion IDM (Chi et al., 2023) uses normalized actions $\bar{\mathbf{A}}_t$. At step q, define $\mathbf{A}_t^q = \sqrt{\bar{\alpha}_q}\,\bar{\mathbf{A}}_t + \sqrt{1-\bar{\alpha}_q}\,\boldsymbol{\epsilon}$ and $\mathbf{v}_q = \sqrt{\bar{\alpha}_q}\,\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_q}\,\bar{\mathbf{A}}_t$. We minimize

$$\mathcal{L}_{\mathrm{IDM}} = \mathbb{E}\left[\sum_{k=0}^{H_a-1} \left\|\mathbf{v}_{q,k} - \mathbf{v}_{\theta,k}(\mathbf{A}_t^q, q; \mathbf{I}_t, \widetilde{\mathbf{g}}_t, \mathbf{w}_{t:t+k})\right\|_2^2\right]. \tag{3}$$

Equation 3 is written token-wise to expose the feedback available for action offset k. The Transformer evaluates all action tokens in parallel using the causal structure described next. Exact token dimensions, network depth, and diffusion schedule appear in Appendix D.

Future-conditioning augmentation. To reduce the IDM's sensitivity to imperfect ForceWM predictions at deployment, we apply random dropout and perturbations of the recorded visual-wrench future. We pair the augmented condition $\widetilde{\mathbf{g}}_t$ with the original action targets (Appendix D).

3.4 Slow-fast execution of ForceWAM

ForceWM updates slowly, but the IDM can rapidly adjust actions with new wrench feedback. Thus, we use a temporally causal structure to enable streaming IDM inference. Furthermore, the slow-fast executor coordinates periodic ForceWM predictions with these reactive action updates.

Temporally causal IDM. The IDM separates conditions by update rate. For a training window, slow encoders map the current observation and regularized recorded future to fixed tokens, while a recurrent fast encoder represents the wrench prefix available at action offset k, $\widetilde{\mathbf{C}}_t^{\mathrm{slow}} = E_{\mathrm{slow}}(\mathbf{I}_t, \mathbf{w}_t, \widetilde{\mathbf{g}}_t)$, $\mathbf{C}_{t,k}^{\mathrm{fast}} = E_{\mathrm{fast}}(\mathbf{w}_{t:t+k})$. The current image is encoded by a convolutional visual backbone, the current wrench by a linear adapter, a separate convolutional adapter encodes the future visual latent, and linear adapters encode the future wrench samples. A unidirectional recurrent encoder produces one fast token per measured wrench, enabling streaming action prediction as new feedback arrives.

Training processes the full synchronized action and wrench window in parallel, but applies lower-triangular masks to both action self-attention and action-to-measured-wrench attention. Action token k accesses only noisy action tokens $0{:}k$ and measured wrench tokens $0{:}k$, ensuring that its inputs are available during streaming inference. A complementary mask hides reference-wrench tokens at window offsets $\leq k$, leaving only later reference samples visible to that action token. This reduces the training-to-inference shift when generated references replace recorded futures. Future visual tokens remain globally visible. These masks implement the token-wise condition in Equation 3 without future-feedback leakage.

ForceWM-IDM slow-fast execution. After training, the recorded future is replaced by a ForceWM prediction. In the synchronous inference process, let r denote the sensor request that starts a slow update and s the control index at which its first new command becomes active. ForceWM generates $\widehat{\mathbf{g}}_r$ from the request-time image, wrench, and task instruction. Because its representation matches the recorded future used for IDM training, it passes through the same slow encoders. Besides, we sample the IDM's initial diffusion-noise tensor,

$$\widehat{\mathbf{g}}_r \sim p_\phi(\mathbf{g}_r \mid \mathbf{I}_r, \mathbf{w}_r, \ell), \quad \widehat{\mathbf{C}}_r^{\mathrm{slow}} = E_{\mathrm{slow}}(\mathbf{I}_r, \mathbf{w}_r, \widehat{\mathbf{g}}_r), \quad \boldsymbol{\xi}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

The generated reference, slow tokens, diffusion noise, and base TCP pose $\mathbf{T}_r$ are cached until the next ForceWM update. At action offset k, reference-wrench visibility follows the same window-indexed mask. With the current feedback prefix $\mathbf{h}_{r,s}^{(k)} = (\mathbf{w}_r, \mathbf{w}_{s+1}, \ldots, \mathbf{w}_{s+k})$, the IDM deterministically re-denoises from the same cached noise:

$$\widehat{\mathbf{a}}_{s,k} = \left[F_\theta\big(\boldsymbol{\xi}_r; \widehat{\mathbf{C}}_r^{\mathrm{slow}}, E_{\mathrm{fast}}(\mathbf{h}_{r,s}^{(k)})\big)\right]_k, \qquad \mathbf{u}_{s+k} = \mathrm{Decode}_{\mathrm{exec}}(\widehat{\mathbf{a}}_{s,k}; \mathbf{T}_r). \tag{5}$$

Only the current action token is decoded and sent to the low-level controller. The cached base TCP pose $\mathbf{T}_r$ converts the relative Cartesian prediction to an absolute pose-gripper command. Within



Figure 3: **Real-robot comparisons of ForceWAM, RDP, ForceVLA, and FA-WAM.**

an action chunk, reusing $\boldsymbol{\xi}_r$ with deterministic DDIM sampling removes variation caused solely by resampling, while the growing measured-wrench feedback prefix changes the conditioning of the newly predicted action token. Deployment settings and inference times are reported in Appendix E.

4 Experiments

We organize the evaluation around four questions. (Q1) Does the complete system improve real-robot contact-rich manipulation? (Q2) How does IDM execution change as it gains access to more task-agnostic play? (Q3) What visual and contact evolution does ForceWM predict during closed-loop execution? And (Q4) how do visual and wrench future conditions contribute to execution?

4.1 Experimental setup

Tasks. We evaluate three real-robot task families under four contact-rich conditions. *Sliding* drags the end effector along the centerline of a force scale while targeting a normal force of 10 N. Success requires full-path completion and satisfaction of the 5–15 N contact-force criterion. *Box flipping* requires sustained contact and a controlled pushing force to carry the box over its edge, and its disturbed variant (Flip-D) adds an external push during contact. *Hanoi ring placement* seats a held ring onto a 10 mm pole, where contact force decides whether the ring seats or jams. All four conditions are contact-rich and use six-axis wrench feedback. They differ in the kind of contact they demand—continuous force maintenance, sustained pushing, rejection of an external disturbance, and fine seating contact.

Datasets. We collect 200 task demonstrations and 200 play episodes for each of the three task families. Flip and Flip-D share the same training data. Both sources provide synchronized wrist RGB, six-axis TCP wrench measurements, and robot actions. Task demonstrations record successful

task executions. Play is collected without an evaluation-task instruction and consists of exploratory pressing, sliding, and making and breaking contact. These recorded visual-wrench transitions provide additional supervision for IDM training.

Baselines. We compare against force-reactive policies RDP (Xue et al., 2025) and ImplicitRDP (Chen et al., 2026), along with ForceVLA (Yu et al., 2025). We also include world-action models Cosmos Policy (Kim et al., 2026), Fast-WAM (Yuan et al., 2026), and FA-WAM (He et al., 2026), as well as $\pi_{0.5}$ (Black et al., 2025). In Table 1, unmarked rows use task demonstrations without action-labeled play, and +Play marks the variants additionally given play. For the +Play variants of ImplicitRDP and Cosmos Policy, task demonstrations are augmented with play trajectories. Cosmos Policy uses an empty language condition for play and learns to predict the recorded actions.

Evaluation. Each evaluation run consists of 20 trials per task from matched initial states, under the same safety limits and controller interface. A trial succeeds if the task completes without a safety stop or human intervention. For Sliding, success additionally requires full-path completion and satisfying the 5–15 N contact-force criterion. We report mean success rates.

4.2 Q1: Overall real-robot performance

We compare ForceWAM with the baselines across the four real-robot conditions to assess task completion under different contact demands. Table 1 summarizes success rates.

Table 1: **Mean real-robot success rates** (the best per column in bold). Slide requires full-path completion with 5–15 N contact force. "+Play": training with additional action-labeled play data.

Method	Slide ↑	Flip ↑	Flip-D ↑	Hanoi ↑
RDP (2025)	15%	40%	30%	45%
ImplicitRDP (2026)	20%	45%	25%	55%
ImplicitRDP (2026) + Play	10%	25%	10%	30%
ForceVLA (2025)	15%	55%	30%	60%
Cosmos Policy (2026)	15%	40%	30%	60%
Cosmos Policy (2026) + Play	10%	20%	15%	40%
Fast-WAM (2026)	10%	50%	25%	65%
FA-WAM (2026)	20%	60%	40%	50%
$\pi_{0.5}$ (2025)	10%	40%	20%	60%
ForceWAM (Task-only IDM)	40%	75%	65%	75%
ForceWAM (Full-play IDM)	**85%**	**95%**	**85%**	**90%**

ForceWAM achieves the highest mean success in all four conditions, and its Task-only IDM also outperforms every external baseline. Force-regulated sliding shows the largest gap. The Full-play IDM reaches 85% success. ImplicitRDP and FA-WAM reach 20%, and the other task-only baselines reach 10–15%. A visual world-action model such as Cosmos Policy can predict plausible motion, but the visual path alone does not specify how to regulate contact force along it. Reactive feedback in RDP and ImplicitRDP, and force-aware fusion in ForceVLA, expose the contact state but do not by themselves learn a stronger goal-conditioned action response to force changes. Because these methods rely on successful demonstrations, they may not observe enough variation in such response behavior outside nominal contact conditions. ForceWAM combines a task-conditioned wrench future with an IDM trained on task and play, giving execution a contact reference and broader action-force experience. The Sliding sequences in Figure 3 illustrate this contact demand: RDP and FA-WAM lose contact before completing the slide, ForceVLA stalls near its initial position, and the shown ForceWAM rollout maintains contact throughout the path and completes the task.

For box flipping, ForceWAM reaches 95% success without disturbances and 85% with external pushes, compared with 60% and 40% for FA-WAM, the strongest external comparator in both conditions. In Figure 3, ForceWAM retains tool-box contact through nominal and disturbed rotation, while the baseline cases lose contact before completing the flip. An external push can change the box-tool relationship, so repeating a nominal motion need not preserve the contact needed for rotation. ForceWM supplies the anticipated box motion and loading, while the IDM updates actions using measured wrench feedback. This provides a mechanism for retaining the flipping objective while adapting local execution, consistent with the higher success in both conditions.

Hanoi ring placement yields a different baseline ordering: Fast-WAM reaches 65%, and Cosmos Policy, ForceVLA, and $\pi_{0.5}$ reach 60%, compared with 90% for ForceWAM. Their relatively strong placement results suggest that visual alignment provides useful guidance here. Alignment alone, however, does not determine whether the ring seats under contact. Conditioning the IDM on the future seating geometry and accompanying wrench evolution gives local force responses a specific placement outcome to pursue. In the Hanoi comparison (Figure 3), ForceWAM adjusts the ring pose and seats it on the pole. All three baseline examples fail to complete placement.

4.3 Q2: Does added task-agnostic play improve execution?

We train five IDMs with all task demonstrations and 0, 50, 100, 150, or 200 play episodes per task family, drawing the intermediate subsets independently (Figure 4). ForceWM and the number of optimizer updates are fixed across variants.

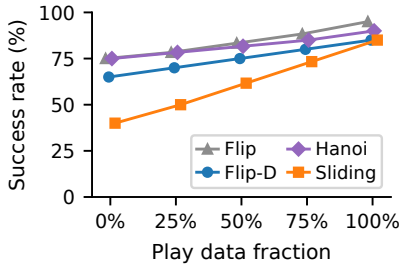

Figure 4: **Mean success with increasing play data.** Task demos, the ForceWM model, and IDM optimizer steps are fixed.

Force-regulated sliding shows the largest absolute gain, from 40% to 85% success in completing the path under the 5–15 N contact-force criterion. This task repeatedly couples motion along a path with force regulation. Successful demonstrations need not cover how to act when contact loads depart from nominal conditions. Pressing, sliding, and release in play provide additional action-wrench transitions for learning such responses. The IDM's future condition associates these actions with their contact outcomes. The large improvement is consistent with the value of this supervision for sustained force regulation.

For box flipping, play raises success from 75% to 95% without disturbances and from 65% to 85% with external pushes. Here, additional experience of building and releasing contact can help the IDM adapt pushing actions as the box rotates. Hanoi ring placement improves from 75% to 90%: local contact experience can likewise help select actions during seating, while the task-trained future retains the required ring-pole geometry. These results suggest that the benefit of play extends across different contact demands, from continuous force regulation to rotation and placement. With ForceWM held fixed, these gains reflect improved execution learning from recorded play futures and actions, without additional task demonstrations.

By contrast, the evaluated end-to-end models, ImplicitRDP and Cosmos Policy, do not benefit from added play data: their shared policies learn directly from play actions even when those actions need not advance the task. The added play can therefore shift their action distributions away from successful task behavior, including when Cosmos Policy also predicts futures. By decoupling knowledge of task-compatible outcomes from knowledge of local action-contact responses, ForceWAM can learn from play in the IDM without training ForceWM on off-task behavior.

4.4 Q3: Visualizing ForceWM predictions during closed-loop execution

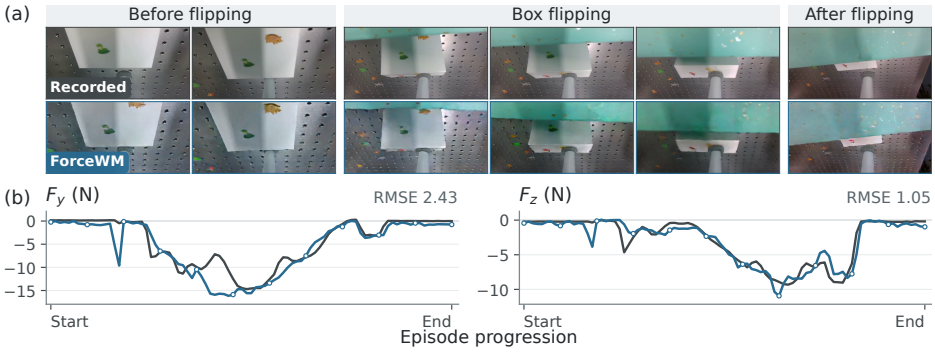

Figure 5: **ForceWM predictions during closed-loop box flipping. (a)** Recorded wrist images (top) and decoded futures (bottom). The three central predictions share one conditioning observation. **(b)** Recorded (gray) and predicted (blue) F_y and F_z (all six wrench components in Appendix F).

We visualize ForceWM predictions recorded during a closed-loop box-flipping rollout on the real robot. Figure 5 compares decoded visual futures and predicted F_y and F_z with the wrist-camera observations and force measurements recorded during execution.

Across the rollout, the visual futures capture the progression from approach through box rotation to separation. During rotation, successive predicted frames preserve the changing orientation of the box relative to the tool, representing the intermediate geometry of the manipulation. The force futures capture the contact loading associated with this motion: load builds as the tool engages the box, remains substantial during rotation, and subsides as the tool separates. Despite discrepancies in local visual details and peak wrench values, the predictions preserve the main interaction stages. This example illustrates how the generated future combines the anticipated tool-object geometry with the accompanying contact loads to guide the IDM.

4.5 Q4: Utility of visual-wrench future conditioning

Table 2 evaluates the utility of visual and wrench future conditioning for online execution. We independently train four shared multi-task IDMs with visual+wrench futures, visual futures only, wrench futures only, or no future conditioning. All four retain the current wrist image and measured wrench feedback and use the full task+play corpus. The visual+wrench row is the Full-play IDM of Table 1. Each future modality is either retained or removed during both training and deployment. The comparison with no future conditioning tests the value of adding generated futures to current observations. The single-modality variants assess the contributions of the visual and wrench components.

Table 2: **Online future-conditioning ablation: mean success rates.** All IDMs use task+play data, with futures from one fixed ForceWM. Rows 1–4 vary future modalities during training and deployment. The last uses both without training-time future-conditioning augmentation.

Future conditioning	**Slide** ↑	**Flip** ↑	**Flip-D** ↑	**Hanoi** ↑
Visual + wrench	**85%**	**95%**	**85%**	**90%**
Visual only	15%	40%	30%	70%
Wrench only	20%	30%	25%	30%
No future conditioning	15%	30%	35%	50%
Visual + wrench, no augmentation	65%	80%	70%	75%

Joint visual-wrench conditioning achieves the highest mean success rate in all four conditions. Relative to visual-only conditioning, it improves success by 55, 55, 20, and 70 percentage points on Flip, Flip-D, Hanoi, and Sliding, respectively. These gains show the value of including a wrench future alongside the visual future. Conversely, retaining the visual future alongside the wrench future improves success by 60–65 percentage points across conditions. Compared with no future conditioning, the joint condition improves success by 40–70 percentage points. Neither individual future modality consistently improves over no future conditioning. These results support complementary guidance from visual and wrench futures under our training and deployment setup.

Finally, we ablate the future-conditioning augmentation introduced in Section 3.3 by retraining the joint visual-wrench IDM without perturbations to its recorded future inputs (final row). Both variants use ForceWM-generated futures at deployment. Augmentation improves success by 15 percentage points on Flip, Flip-D, and Hanoi, and by 20 points on Sliding. These gains show that regularizing recorded future conditions improves execution with ForceWM-generated futures.

5 Conclusion

We presented ForceWAM, which decouples knowledge of task-compatible interaction outcomes from knowledge of how local actions change contact. Task demonstrations train a visual-wrench future model, while task and play data train an inverse dynamics model that translates generated futures into actions with force feedback. Real-robot experiments show higher task success than the evaluated baselines and consistent gains from play. This suggests that using task and play data to learn complementary knowledge offers a promising path to scalable, stable contact-rich manipulation.

AI USE STATEMENT

Large language models were used to assist with programming and figure preparation, as well as grammar and expression polishing. They did not contribute to research ideation, methodology design, or experiments.

ETHICS STATEMENT

This work complies with the ICLR Code of Ethics. It does not involve human or animal subjects, nor the use of private or sensitive data. The research raises no direct ethical or legal concerns, and the authors are committed to responsible and fair use of the proposed methods.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of our work. The proposed model, implementation details, and evaluation protocols are described in detail in the main paper and appendix. The source code and data will be made publicly available.

REFERENCES

AgiBot Research Team, Renhang Liu, Wenzhi Zhao, Zhuo Yang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Guanghui Ren, Youlun Peng, Rongjun Jin, Nan Wang, Sukai Wang, Xindong He, Jinyuan Feng, Ziyu Xiong, Linqing Zhong, Yifei Wei, Feng Han, Long Zhang, Da Huang, Nanshu Zhao, Chenghao Yin, Mo Wu, Zhaodong Yan, Kongtao Hu, Yuxiang Yan, Aogelijiang Niyazi, Yu Fang, Jia Zeng, Lizhu Meng, Daizhen Lv, Haoyu Cao, Zhiwen Hou, Lianjin Ye, Yuehan Niu, Zhikai Cai, Xuan Hu, Hui Min, Xiongfeng Cai, Yue Liao, Jing Wu, Soujanya Poria, Ye Li, Sanping Zhou, and Maoqing Yao. GE-Act 2.0: Pretraining and scaling a world-action model for robotic manipulation. arXiv preprint arXiv:2609.05588, 2026.

Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Cheston Tan, Yunzhu Li, and Jiajun Wu. RoboPack: Learning tactile-informed dynamics models for dense packing. In *Robotics: Science and Systems*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026a.

Hongzhe Bi, Zihao Zhou, Yihang Tang, Jingrui Pang, Shuhe Huang, Haitian Liu, Runqing Wang, Shuai Huang, Yichen Wang, Yiming Cheng, Ruowen Zhao, Zhenghua Li, Hengkai Tan, Xiaolong Liu, Jinhui Wan, Jiabao Liu, Min Zhao, Fan Bao, and Jun Zhu. Motus2: A self-evolving general world model for dexterous manipulation. arXiv preprint arXiv:2608.30237, 2026b.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of the Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. arXiv preprint arXiv:2606.09811, 2026.

Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. doi: 10.1109/LRA.2018.2852779.

Wendi Chen, Han Xue, Yi Wang, Fangyuan Zhou, Jun Lv, Yang Jin, Shirun Tang, Chuan Wen, and Cewu Lu. ImplicitRDP: An end-to-end visual-force diffusion policy with structural slow-fast learning. *IEEE Robotics and Automation Letters*, 11(8):10010–10017, 2026. doi: 10.1109/LRA.2026.3710031.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, 2023.

Hongjie Fang, Shirun Tang, Mingyu Mei, Haoxiang Qin, Zihao He, Jingjing Chen, Ying Feng, Chenxi Wang, Wanxi Liu, Zaixing He, Cewu Lu, and Shiquan Wang. Force Policy: Learning hybrid force-position control policy under interaction frame for contact-rich manipulation. In *Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.128.

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.

Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models. arXiv preprint arXiv:2602.06001, 2026.

Neville Hogan. Impedance control: An approach to manipulation: Part I—theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 1985. doi: 10.1115/1.3140702.

Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. In *IEEE International Conference on Robotics and Automation*, pp. 4829–4836, 2025. doi: 10.1109/ICRA55743.2025.11128452.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, 2026.

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *IEEE International Conference on Robotics and Automation*, 2019.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems*, 2026a. doi: 10.15607/RSS.2026.XXII.016.

Sizhe Lester Li, Evan Kim, Xingjian Bai, Tong Zhao, Tao Pang, Max Simchowitz, and Vincent Sitzmann. Turning video models into generalist robot policies. arXiv preprint arXiv:2605.27817, 2026b.

Yang Li, Zhaxizhuoma, Hongru Jiang, Junjie Xia, Hongquan Zhang, Jinda Du, Yunsong Zhou, Jia Zeng, Ce Hao, Jieji Ren, Qiaojun Yu, Cewu Lu, Yu Qiao, and Jiangmiao Pang. 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, 2026c.

Wenhai Liu, Junbo Wang, Yiming Wang, Weiming Wang, and Cewu Lu. ForceMimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation*, 2025.

Zipei Ma, Xiaofei Wei, Junzhe Jiang, Shunlin Lu, and Li Zhang. TacPAC: Tactile prediction and real-time action correction in world-action models for contact-rich manipulation. arXiv preprint arXiv:2609.05266, 2026.

Rishabh Madan, Angchen Xie, Samantha Saak, Andres Blanco, Dohyeok Lee, Sarah Grace Brown, Yunting Yan, Mark Zolotas, Jose Barreiros, and Tapomayukh Bhattacharjee. TACTIC: Tactile and vision conditioned contact-centric control for whole-arm manipulation. In *Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.060.

NeoteAI Team and Fudan TEAI Team. N_0-TWAM: Scaling tactile-native world-action model for contact-rich manipulation. arXiv preprint arXiv:2607.23783, 2026.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, et al. Cosmos world foundation model platform for physical AI. arXiv preprint arXiv:2501.03575, 2025.

Quanquan Peng, Yutong Liang, Rui Yan, Nicklas Hansen, and Xiaolong Wang. FACT: Failure-aware causal training for world-action models. arXiv preprint arXiv:2608.10232, 2026.

Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joseph Ortiz, and Mustafa Mukadam. NeuralFeels with neural fields: Visuotactile perception for in-hand manipulation. *Science Robotics*, 9(96):eadl0628, 2024. doi: 10.1126/scirobotics.adl0628.

Yi Wang, Wendi Chen, Zimo Wen, Han Xue, Xueqi Li, Wenye Yu, Zhijie Chen, Hao Yang, Jun Lv, Chuan Wen, and Cewu Lu. Never too late for force: Accelerating VLA post-training with reactive force injection. In *Proceedings of the Conference on Robot Learning*, 2026. Accepted.

Zhengtong Xu, Qiang Qiu, and Yu She. VILP: Imitation learning with latent video planning. *IEEE Robotics and Automation Letters*, 10(4):3350–3357, 2025. doi: 10.1109/LRA.2025.3542317.

Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.052.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Jim Fan, and Joel Jang. World action models are zero-shot policies. arXiv preprint arXiv:2602.15922, 2026.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, Wenqiang Zhang, and Cewu Lu. ForceVLA: Enhancing VLA models with a force-aware MoE for contact-rich manipulation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 93409–93439, 2025. doi: 10.52202/085713-3124.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? arXiv preprint arXiv:2603.16666, 2026.

Yuhang Zheng, Songen Gu, Yupeng Zheng, Weize Li, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, Haoran Li, Yilun Chen, Shuicheng Yan, and Wenchao Ding. OmniVTA: Visuotactile world modeling for contact-rich robotic manipulation. arXiv preprint arXiv:2603.19201, 2026.

Jianyi Zhou, Feiyang Hong, Yunhao Li, Yicheng Zhao, Yongjue Cen, Zirui Liu, Jiakang Huang, Zirui Chen, Ruiyang Zhang, Weizhuo Zhu, Xuhua Song, and Shuo Yang. TouchWorld: A predictive and reactive tactile foundation model for dexterous manipulation. arXiv preprint arXiv:2607.07287, 2026.

Lifeng Zhuo, Wendi Chen, Han Xue, Shirun Tang, Jun Lv, Cewu Lu, and Chuan Wen. FA-RDP: A frequency-adaptive reactive diffusion policy for contact-rich manipulation. arXiv preprint arXiv:2607.28596, 2026.

A HARDWARE SETUP AND DATA COLLECTION

All real-robot experiments use a single Flexiv Rizon 4S 7-DoF collaborative arm equipped with joint torque sensors and a six-axis force/torque (F/T) sensor at the end effector (Figure 6). The end-effector configuration depends on the task. Hanoi ring placement uses a Robotiq 2F-85 parallel-jaw gripper. For box flipping (Flip and Flip-D) and Sliding (Slide), a joystick and custom compliant fingertips are mounted at the end effector.

We collect Hanoi data via virtual-reality (VR) teleoperation using a Meta Quest 3. For box flipping and Sliding, the joint torque sensors support kinesthetic teaching during data collection, while the end-effector F/T sensor records contact forces and torques. Across tasks, this F/T sensor supplies the six-axis wrench observations used by ForceWM and the IDM. A wrist-mounted RGB camera records visual observations. For Hanoi, we also use a third-person camera.

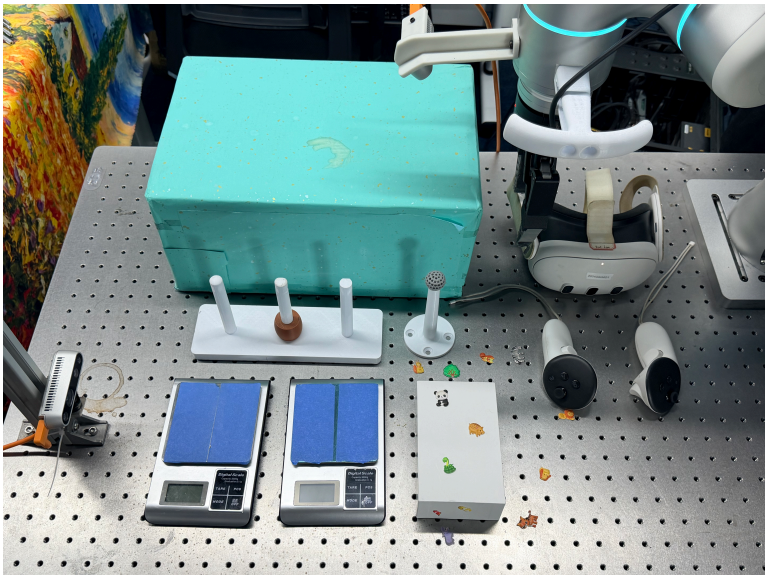

Figure 6: Hardware setup and task objects.

B VISUAL-WRENCH REFERENCE CONSTRUCTION

Recorded training futures and ForceWM predictions use the same representation. From the current image at time t, the visual target samples future wrist frames at offsets $\{4, 8, 12, 16\}$ and encodes them into one $16 \times 56 \times 56$ video latent. The wrench target contains the next 16 six-dimensional TCP wrench measurements. Force and torque components are normalized independently by a min-max map n_F fitted on the ForceWM training split. The same frozen n_F is used for recorded future trajectories and online ForceWM predictions. The measured causal wrench stream supplied to the IDM uses a separate min-max map n_I fitted on its training data. The future-wrench stream remains in the n_F coordinates and passes through its IDM input normalizer unchanged.

ForceWM uses the temporal input convention of its pretrained video tokenizer. Its 17-frame pixel sequence is organized as one blank frame, four copies of a current-wrench placeholder, four copies of the current wrist image, four future-wrench placeholders, and the four sampled future wrist frames. The video tokenizer temporally compresses this sequence into five $16 \times 56 \times 56$ latent slots: blank, current wrench, current image, future wrench, and future visual content. The first three are clamped conditions. Only the last two are noised and predicted.

A fixed adapter flattens the current 6-D wrench or the future 16×6 trajectory and periodically repeats it across the corresponding video-latent-shaped slot. Decoding averages the complete repeats for each component. This parameter-free packing lets the wrench and video slots share the pretrained spatiotemporal backbone while keeping the two generated outputs separate. If a complete future

modality crosses an episode boundary, its generated slot is excluded from the loss rather than supervising on padded values.

C Robot action representation

Each robot action comprises a Cartesian TCP-pose target and a gripper command. In the training data, it is represented by the TCP pose and gripper state at the next recorded timestep. Poses are expressed relative to the base TCP pose at the start of the prediction window. Rotation is represented by roll, pitch, and yaw. At deployment, the predicted relative pose is composed with the cached base TCP pose and converted to the absolute pose-gripper command accepted by the robot controller.

The implementation inherits an auxiliary prediction target from ImplicitRDP (Chen et al., 2026). In addition to the executable relative pose-gripper action, it predicts a force-derived virtual pose and a scalar stiffness. This auxiliary block is trained jointly with the executable block, but is discarded after inference: the robot receives only the decoded TCP pose and gripper width. The auxiliary target regularizes representations learned from the IDM's multimodal inputs. After relative-pose conversion, the full diffusion target has 14 dimensions and an 18-step prediction horizon. Only the prefix aligned with the current slow window is eligible for execution.

D Architecture and training details

ForceWM. We initialize the spatiotemporal backbone from the Cosmos Predict 2B weights (NVIDIA et al., 2025) and train ForceWM directly on task demonstrations. No intermediate task-specific pretraining is used. The model takes a wrist-camera image, a six-dimensional TCP wrench, and a task instruction (e.g., "flip box"). Visual inputs are encoded at 448×448. The five-slot latent is denoised by the shared Cosmos video diffusion backbone with frame-replacement conditioning. A hybrid EDM noise distribution draws 70% of noise scales from the configured log-normal distribution and 30% uniformly from $[1, 85]$. The sigma-weighted MSE is evaluated only on the future-wrench and future-visual slots. The task instruction is supplied through its pretrained T5 text embedding. We train for 20,000 iterations on eight GPUs with batch size 8 per GPU, hold out the final 30 episodes for validation, and save a checkpoint every 1,000 iterations. The optimizer uses learning rate 3×10^{-5}.

IDM. The IDM encodes the current 224×224 wrist image with a ResNet-34 whose normalization layers are GroupNorm. The resulting 7×7 feature map is projected into 49 tokens of width 768, and the current 6-D wrench is linearly projected into one additional token. The $16 \times 56 \times 56$ future visual latent is encoded by $\mathrm{Conv}_{3\times3}(16, 256)$-GroupNorm-SiLU, pooled to a 2×2 map, and linearly projected into four 768-D tokens. Each of the 16 future 6-D wrench samples is independently projected into one 768-D token. A one-layer, unidirectional GRU with hidden size 512 followed by a linear projection produces a 768-D token for every measured wrench in the causal prefix.

The action denoiser linearly embeds each 14-D noisy action and adds learned temporal positions. The diffusion timestep uses a sinusoidal embedding. A learned projection and a two-layer Mish MLP encode the fixed condition memory. A seven-layer pre-normalized Transformer decoder of width 768, four attention heads, and $4\times$ feed-forward expansion predicts velocity for all 18 action tokens during training. Its action self-attention mask and its cross-attention mask over GRU wrench tokens are both lower triangular. A complementary mask hides every future-wrench reference token whose window offset is less than or equal to the queried action token. Therefore action token j can use noisy-action and measured-wrench tokens $0{:}j$, the current image and wrench, reference-wrench tokens at later window offsets, and all fixed future-visual tokens. Processing the full window in one forward pass is thus parallel teacher-forced training with the same causal information boundary used by streaming inference, rather than access to future measured feedback. The action horizon is 18 and the model observes one current image. Deployment refreshes the slow reference after six actions, so only tokens $0{:}5$ are eligible for execution. In particular, the two tokens beyond the 16-step reference horizon are supervision-only.

For the play-scaling study, we sample training windows at random from the pooled task and selected play trajectories. The action model uses a 100-step squared-cosine diffusion schedule and velocity prediction. Training adds $\boldsymbol{\epsilon} + 0.1\boldsymbol{\epsilon}'$ to the normalized action while computing the velocity target from $\boldsymbol{\epsilon}$. Inference uses five deterministic DDIM steps with $\eta = 0$, leading timesteps $[80, 60, 40, 20, 0]$, and

clips the predicted normalized clean action to $[-1, 1]$ before denormalization. Image augmentation consists of a center crop, rotation in $[-5^\circ, 5^\circ]$, and color jitter during training. Stochastic image augmentation is disabled online.

To reduce the recorded-to-predicted future shift, IDM training applies future-conditioning augmentation. It first selects the temporal lookup, then drops the entire future condition with a fixed probability, followed by independent modality dropout, additive noise, and multiplicative scale jitter for the surviving visual and wrench parts. The exact flip-box values are summarized in Table 3.

Table 3: Future-conditioning augmentation used for the flip-box IDM.

Augmentation	**Flip-box setting**
Whole-condition dropout	0.10
Visual / wrench dropout	0.25 / 0.50
Scale jitter	$[0.85, 1.15]$
Wrench noise std.	0.075 in normalized coordinates
Visual-latent noise std.	0.075 times latent standard deviation
Temporal offsets	$\{0, -1, -2, -3, -4, -5, -6, -9, -12\}$

E DEPLOYMENT DETAILS

We deploy ForceWM and the IDM in a shared policy service. ForceWM uses one denoising step and the IDM uses five. Measured separately on the deployment system, their calls take approximately 100 ms and 60 ms, respectively. The reference is refreshed every six actions.

Each refresh caches the generated visual-wrench reference, slow tokens, initial IDM noise, and base TCP pose used for relative-to-absolute action conversion. Between refreshes, the IDM deterministically re-denoises the action prefix from cached noise using the growing measured-wrench prefix, executing only its current action token. The GRU re-encodes the available wrench prefix at each call, and all reference-specific caches reset at the next refresh.

At each reference refresh, ForceWM runs first, followed by the IDM. The previous controller target remains active during these calls. The first action after a refresh uses the request-time wrench, and subsequent actions incorporate new measurements. Reference-wrench visibility follows the action index within the window.

F FULL SIX-AXIS WRENCH PREDICTION COMPARISON

Figure 7 extends the F_y and F_z visualization in Figure 5 to all six wrench components from the same closed-loop box-flipping rollout.

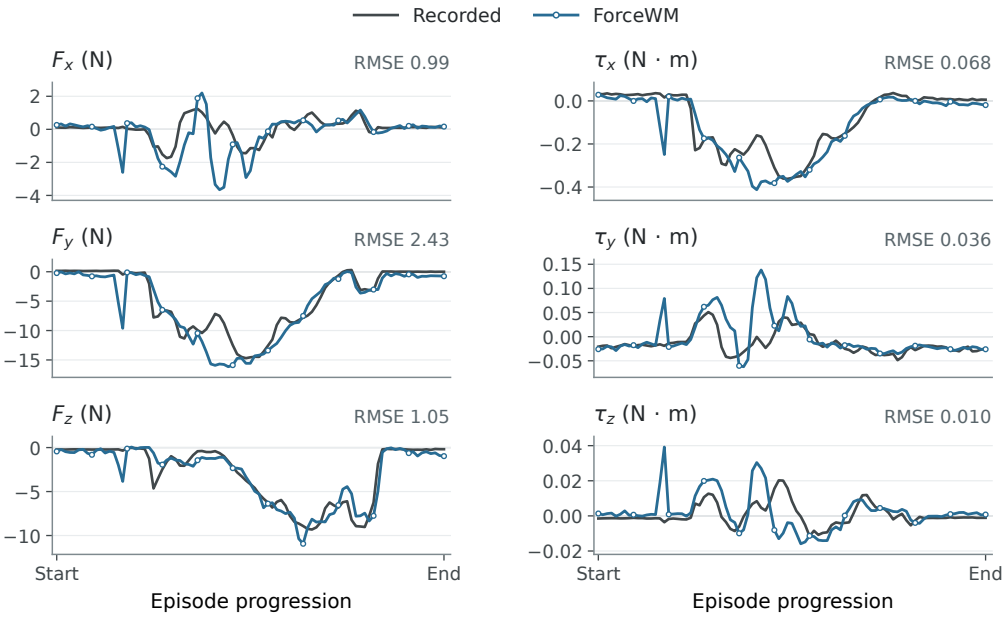

Figure 7: Full six-axis comparison for the rollout in Figure 5. Recorded (gray) and predicted (blue) forces are shown on the left and torques on the right.

Beyond the force profiles in the main paper, the predicted τ_x also captures the overall torque buildup and release during box flipping. Together with the visual futures in Figure 5, these predictions describe the evolving tool-object geometry and the accompanying interaction loads, providing complementary references for the IDM.